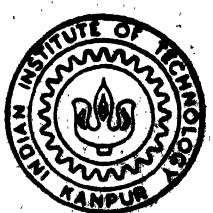
ANUSARAKA: A DEVICE TO OVERCOME THE LANGUAGE BARRIER

by

V. N. NARAYANA



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

January, 1994

ANUSARAKA: A DEVICE TO OVERCOME THE LANGUAGE BARRIER

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

by

V. N. Narayana

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1994

CERTIFICATE

It is certified that the work contained in this thesis entitled "ANUSARAKA: A DEVICE TO OVERCOME THE LANGUAGE BARRIER", by V. N. Narayana, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Prof. Rajeev Sangal
Thesis Supervisor
Department of Computer Science and Engg.
Indian Institute of Technology, Kanpur-208016
India

January, 1994.

SYNOPSIS

The problem addressed in this thesis is: how to overcome the language barrier with the help of machine. This question is explored for Indian languages in general, and Kannada-Hindi, in particular. The language barrier to be overcome relates to written text available in machine readable form.

It should be emphasized that the problem posed is different from the Artificial Intelligence problem. The latter is how to build an intelligent machine which can understand and generate natural language (like a human being). The present problem is how to let a human being read or access written text in a language he does not know. Here the human being has to understand the text, whereas the AI problem is for the machine to understand the text.

The practical aspect of this problem is to divide the load between man and machine in such a way that the aspects which are difficult for the human being are handled by the machine, and aspects which are easy for the human being are left to him. The answer lies in separating language based analysis from knowledge and inference based analysis. The former task is taken up by the machine, while the latter is left to the human reader.

The approach, however, cannot be a brute force method because the problem is very complex. A brute force solution will either blow up in the large amount of resources it will need, or the large amount of time it will take. Another desirable feature of the approach would be the ease with which the system can be extended to other related languages. The system is likely to possess these qualities only if it is based on sound principles and theory.

Language based analysis primarily requires knowledge of grammar and lexicon.

Some of the coding of information is quite explicit in grammar and does not require large amount of processing. For example, the post-position marker 'ne' in Hindi specifies the karta relation between a verb and a noun. This coding is simple and the extraction of this information by a processor is immediate. On the other hand, certain information is coded deeply in grammar. It is not readily available without a lot of processing. An example is the 'kara' (having-done) suffix for verbs in Hindi. Such verbs do not have an explicit nominal as their karta, instead the karta is shared with another verb that such a verb modifies. Using this sharing rule to extract karta relation, however, requires first identifying what verb is modified and what is its karta.

The lexicon too has similar coding. Each word has a nuclear sense which can be specified (and learnt) easily, however, what senses a word spans is part of a much more elaborate coding involving pragmatic issues.

Anusaraka uses the above language based knowledge to analyse a given input string and to extract information from it. This information is presented in a language close to the target language. The reader by making use of his background knowledge should be able to get the "intended" meaning in the original input string.

The output language is close to the target language, but has a syntax of its own. It may even be called a dialect of the target language. The reader will usually require some learning of the dialect. But this learning time will be negligible when compared to the learning time of the source language.

The main problem of anusaraka is how to present the information extracted from the source text in the target language. As it does not use any background knowledge, it may have to take an incomplete "picture" obtained from the source and encode it in the target language. There is a problem in coding exactly the same information from one language to another. This problem arises because we want to generate a sentence of about equal length

and paralleling the sentence construction wherever possible. Clearly, it may be possible to express the same information by a longer prose, but if the size is much bigger than the original, it makes it difficult for the reader to get the same meaning as in original source language. Flavour of the original is also lost.

The anusaraka answer is to deviate from the target language in a systematic manner. First, new notation is invented and incorporated. Thus, we can decide to have ko' to mean dative case marker in Hindi as distinct from accusative marker (ko). Second, we may relax some of the conditions in the target language. For example, we might give up agreement in our "dialect" of the target language. Some of the constructions of the source language may also get introduced in the target language. Existing words in the target language may be given wider or narrower meaning.

Sometimes the syntactic differences between languages can be overcome by building language bridges. For Kannada-Hindi anusaraka for example, the major syntactic differences can be bridged by a few additional functional particles or suffixes. The most striking among them is the "jo" construction for handling adjectival participial phrases in Kannada. Kannada has a large number of adjectival participial forms for verbs, and they code tense-aspect-modality information. Hindi, on the other hand, has only two adjectival participles covering just the perfective and present continuous. Thus, there is a syntactic hole which creates difficulty for anusaraka. The difficulty is solved by mapping the Kannada participial phrase to relative clause ("jo" construction) in Hindi which permits tense aspect modality information to be coded. This, however, creates another problem. The relative clause in Hindi requires case marking or postposition for the nominals in the clause. This information is not coded in the original Kannada phrase. This absence of information in the original string is shown by marking the vibhakti by '*' in the anusaraka Hindi.

Catastrophe occurs in anusaraka output when reader either fails to comprehend

the meaning or misinterprets the meaning. Several catastrophe avoidance strategies are discussed including the training of the user and evolving a rich notation for the anusaraka output language. It is important to build an intelligent user interface which can assist the user and avoid catastrophe whenever necessary.

Theoretical issues of interest relate to information and how it is coded in language. How is the information extracted? What are the sources of knowledge that are used in information extraction? How should the knowledge be organized? etc. Future potential and work is also discussed.

A complete anusaraka system for Kannada-Hindi has been built based on above ideas, with a Kannada vocabulary of 30,000 root words. It has been tested on a large number of texts taken from different domains. Its implementation is described together with sample outputs.

ACKNOWLEDGEMENTS

With great pleasure and deep sense of gratitude, I acknowledge the invaluable guidance of Prof. Rajeev Sangal, my thesis supervisor. In spite of his extremely busy schedule, he could find time to provide precious guidance and that extra bit of support, required for the completion of any thesis. His never ending patience and ever smiling face was a source of inspiration for me to complete this work.

I express my deep sense of indebtedness to Dr. Vineet Chaitanya for the continued support he has provided throughout this work. He is the main person who is responsible for the birth of *anusaraka*. He was always there to help me and put me on the right track which led to the smooth finish of my thesis. His constructive criticisms have had a great bearing on the form and content of this thesis.

I express my sincere gratitude to Prof. B. N. Patnaik for his keen interest in my work and constant support given during my stay at IIT/K.

My sincere thanks are due, to The Principal, Malnad College of Engineering, Hassan and The Director of Technical Education in Karnataka for deputing me to IIT, Kanpur to pursue my higher studies.

Many people have helped me in carrying out my work smoothly. I express my thanks to Mrs. Amba Kulkarni, Mr. Vasudev Varma, Mr. Navdeep Sood, Ms. Aditi Agarwal and Ms. Chinmoyee for providing the programming support. I thank Dr. Umamaheshwara Rao, Dr. Meena Belliappa, Dr. Subbukrishna, and Dr. K.V. Ramakrishnamacharyulu for providing the linguistic support. I thank Mrs. Kamalagangadaraiah and students from Karnataka of IIT/K for acting as Kannada native language specialists. I owe special thanks to the members of Akshar Bharati for teaching me Hindi and for becoming Hindi native language consultants.

It was great to have persons like Prof. Subramanya, Mrs Prabha Subramanya and Mrs Leela (Mataji) who never let me feel that I was away from my parents.

I acknowledge the support received from my friends Sheeni and Rashi during the initial stage of my stay at IIT/K and NVB Rao, Dr. Jha and Deepak Gupta during the final stage of my thesis work.

Last, but not the least, I would come to the persons who suffered due to my Ph.D. I owe an excuse to my parents, my wife, Gayathri and my daughter, Divya, for not giving proper attention to their needs during my thesis work. I thank my wife for her emotional support and constant encouragement during the course of this work.

Contents

List of Figures

NOTATIONS USED IN THE THESIS

K : Kannada sentence

H : Hindi sentence

E : English sentence

ṀH : Anusaraka Hindi sentence

!H : Hindi gloss

* : An asterisk preceding a sentence indicates that the sentence is ungrammatical

abl. : ablative

acc. : accusative

dat. : dative

emph. : emphatic

erg. : ergative

f. : feminine

fut. : future

inst. : instrumental

m. : masculine

nom. : nominative

past_perf. : past perfective

ppl. : past participle

pres. : present

Chapter 1

Introduction

1.1 Background

Natural Language Processing (NLP) has emerged as an important area in recent times. It deals with computational models of natural language analysis, generation, etc. It is important for several reasons:

- NLP can be used to build systems that overcome language barriers among people. For example, machine translation systems can be developed based on advances made in NLP.

- NLP can make it possible for common people to communicate with the computer without learning specialized languages and packages. If computers have to be accessible to vast numbers of people, then NL capability in the computer is essential. It should be noted that knowledge and technology of NLP related to Indian languages is not likely to be available from any part of the outside world. It has to be developed by us here.

- NLP is at the cutting edge of research involving several disciplines, for example, Computer Science and Artificial Intelligence, Linguistics and Logic, etc. It is expected

in turn to make a major impact on several of these disciplines.

The problem being addressed in this thesis is how to overcome the language barrier in India. Fully-automatic general-purpose high-quality machine translation systems (FGH-MT for short) are extremely difficult to build. There are no existing translation systems for any pair of languages in the world that qualify to be called FGH-MT. The difficulty arises from the following reasons:

1. In any natural language text only part of the information to be conveyed is explicitly expressed and it is the human mind which fills up the details by using world knowledge. The basic reason for this state of affairs is: The concepts and shades which a natural language purports to describe form a continuum and the number of lexical items available are finite, so necessarily they have to be overloaded and it is only because of the total context and shared background understanding that we are able to disambiguate them and are able to communicate.

2. Different natural languages adopt different conventions about the type and amount of information to be used. The reasons for this may be: the history of language development (differing tastes, arbitrary choices made in the long history of language, presence of other languages and mixing with them, etc.), the envisaged primary function of the language, etc.

The net result of this is that unless we provide machines with knowledge and inference capability comparable to that of human beings, FGH-MT will not be feasible. It will not be an exaggeration to say that in spite of tremendous progress in computer technology (mainly in terms of speed, memory and programming environments) FGH-MT remains a distant dream.

1.2 Cutting the Gordian Knot

In spite of the difficulty of FGH-MT, it can be claimed that with the help of machines, language barrier can be overcome today. If this sounds paradoxical let us consider an analogy: Scientists even today are struggling to build a machine that can walk like a human, avoiding obstacles to reach a destination. At the same time, we have been successfully using rail transport for more than a century. The distance barrier has been overcome even without the machines learning to walk.

True, for this we have had to lay railway tracks all over the country; build bridges across the rivers; dig tunnels through mountains and build a huge infra-structure to make the whole thing feasible. Even then it delivers the goods and passengers at only the railway stations; we need separate arrangements to get the things and persons at home! But all said and done it does enable us to overcome the distance barrier.

If FGH-MT is like the walking machine, what is the counterpart to the railway locomotive? The answer lies in separating language-based analysis of text, from knowledge and inference-based analysis. The former task is left to the machine, and the latter task to the human reader because of their proficiency for the respective tasks. We also relax the requirement that the output be grammatical. We do require, however, that the output be comprehensible. Apart from the effort to build appropriate multilingual databases from computational viewpoint, creative ideas are needed to establish languages bridges. Though this is challenging, it is definitely feasible. With the appropriate infra-structure the language barrier can be overcome with today's technology at least among the Indian languages.

1.3 Usability

Usability is a new concept that says that a product should be designed to involve the user at various levels of its operation. Rather than viewing the user as "just a user" who should not need to delve into the working of the machine, usability plans for his intervention if needed (See Adler et al. (1992)). Take the example of the xerox machine. When everything is going fine, the user simply presses a button and out comes a photocopy. But suppose the paper gets jammed. Now the user should be allowed (even encouraged) to play the role of someone who can fix the problem. This is done by allowing for a simple method of opening the machine. After the machine is opened, the necessary parts should be easily accessible. They should even be colour coded to facilitate fixing of the problem. Or alternatively, the machine may run out of paper or toner. The same concerns apply. The best way to answer these concerns is to plan for them at the design stage itself.

Usability brings about two fundamental changes in viewpoint. First, it accepts that it is not possible to design for successful operation of machine hundred percent of the time. Second, it considers user as a capable and willing problem solver, and not just a passive receiver of a service.[1]

While designing a device as complex as one that can overcome the language barrier, incorporation of usability is inevitable. At a very gross level, we have already said in the last section, the load has to be shared between man and machine. But usability requires more than just that. To use the analogy with xerox machine, what should be done when "a word or a sentence gets jammed" in a translation machine? The same analogy provides us the answer: "the user should be able to open the translation machine, and use his ingenuity to remove the word that is stuck". This requires that the working of the translation machine be kept simple and transparent. Transparency will help the user in identifying the cause

[1]There are issues relating to safety, but they do not concern us here.

of failure. Moreover, there should be sufficient "colour coding" in terms of help files, user interfaces etc., so that the user can interpret or guess the meaning of the word that got stuck. He should be able to seek help from the machine.

Note that transparency (in the context of usability) says that if there is a linguistic rule that improves the performance from 80 to 90 percent, say, but hides or obscures the working of the system, do not adopt it. Interestingly, in the history of xerox machines when paper jamming was a serious problem, there were design proposals for very elaborate paper feeding mechanisms which would reduce paper jams but were very complex and "inaccessible" to the user. These were rejected in favour of a simple mechanism that did not avoid the jam, but allowed the user to clear it quickly.[2]

1.4 Symbiosis

There is another consideration which is very important while building a machine to overcome the language barrier[3]: How to combine the theory and practice appropriately. The theory might have some general principles that cover a large number of cases but which work only under "ideal" conditions (which actually do not occur in real life). The practice, on the other hand, has a large number of special rules each of which work in the real life but only for few cases. Alone, each of these two chunks of knowledge (theoretical knowledge and practical knowledge) are not sufficient to solve the broad real life problems. Therefore, how should these two diverse chunks be combined?

The symbiosis between theory and practice while solving real life problems guides both in what they should address. It identifies what theoretical problems are of interest. How can the complexity or number of special cases be controlled? What concepts and cate-

[2]It is also interesting to note that the problem got solved later partly because of better quality more uniformly thick paper, much like the future availability of better quality dictionaries which might reduce word jams in a translation machine.

[3]Or while building any machine for that matter.

gories are important? And similarly for practice: what special cases (which are not covered by theory) to work within seeking thumb rules, gathering data, conducting experiments, etc.

There is another important facet of the symbiosis while solving real life problems. It tends to cross disciplinary boundaries. Problems do not come neatly categorized in different disciplines. In solving them, one needs to draw knowledge from several sources. For example, in working on machine translation, it might require knowledge from diverse sources, such as: linguistics, computer science, vyakarana (traditional grammar), etc.

The real life also helps to cut the problem differently. For example, machine translation systems should be designed keeping in mind the need. In case of India, there is an obvious need to translate among Indian languages. This can spark off new research leading to universals that hold among Indian languages.

1.5 The Problem

The problem addressed in this thesis is: how to overcome the language barrier with the help of machine. This question is explored for Indian languages in general, and Kannada-Hindi, in particular. The language barrier to be overcome relates to written text available in machine readable form.

It should be emphasized that the problem posed is different from the Artificial Intelligence problem. The latter problem is how to build an intelligent machine which can understand and generate natural language (like a human being). The present problem is how to let a human being read or access written text in a language he does not know. Here the human being has to understand the text, whereas the AI problem is for the machine to understand the text.

The practical aspect of this problem is to divide the load between man and machine

in such a way that the aspects which are difficult for the human being are handled by the machine, and aspects which are easy for the human being are left to him. The aim is to minimize the effort of the human being.

The approach, however, cannot be a brute force method because the problem is very complex. A brute force solution will either blow up in the large amount of resources it will need, or the large amount of time it will take. Another desirable feature of the approach would be the ease with which the system can be extended to other related languages. The system is likely to possess efficiency, extensibility, etc. only if it is based on sound principles and theory.

Theoretical issues of interest relate to information and how it is coded in language. How is the information extracted? A related question is why the same amount of surface information in one language is clearly understood while in another language, it is not clear to the reader. What are the sources of knowledge that are used in information extraction? How should the knowledge be organized?

1.6 Achievements and Contributions

This thesis solves the problem posed above. It describes a system called anusaraka that can overcome the language barrier. Such a system has been built for Kannada-Hindi. Given a Kannada text, it produces output in anusaraka Hindi. The latter is close to Hindi and it may be called a dialect of Hindi, since it has a grammar of its own. The reader can learn this new language relatively easily.

The main problem of anusaraka is how to present the information extracted from the source text in the target language. As it does not use any background knowledge, it may have to take an incomplete "picture" obtained from the source and recode it in the target language. There is a problem in coding exactly the same information from one language to

another. This problem arises because we want to generate a sentence of about equal length and paralleling the sentence construction wherever possible. Clearly, it may be possible to express the same information by a longer prose, but if the size is much bigger than the original, it makes it difficult for the reader to get the same meaning as in original source language. Flavour of the original is also lost.

The anusaraka answer is to deviate from the target language in a systematic manner. First, new notation is invented and incorporated. Thus, we can decide to have ko' to mean dative case marker in Hindi as distinct from accusative marker (ko). Second, we may relax some of the conditions in the target language. For example, we might give up agreement in our "dialect" of the target language. Some of the constructions of the source language may also get introduced in the target language. Existing words in the target language may be given wider or narrower meaning.

Sometimes the syntactic differences between languages can be overcome by building language bridges. For Kannada-Hindi anusaraka for example, the major syntactic differences can be bridged by a few additional functional particles or suffixes. The most striking among them is the "jo" construction for handling adjectival participial phrases in Kannada. Kannada has a large number of adjectival participial forms for verbs, and they code tense-aspect-modality information. Hindi, on the other hand, has only two adjectival participles covering just the perfective and present continuous. Thus, there is a syntactic hole which creates difficulty for anusaraka. The difficulty is solved by mapping the Kannada participial phrase to relative clause ("jo" construction) in Hindi which permits tense aspect modality information to be coded. This, however, creates another problem. The relative clause in Hindi requires case marking or postposition for the nominals in the clause. This information is not coded in the original Kannada phrase. This absence of information in the original string is shown by marking the vibhakti by '*' in the anusaraka Hindi.

Catastrophe occurs in anusaraka output when reader either fails to comprehend the meaning or misinterprets the meaning. Several catastrophe avoidance strategies are discussed including the training of the user and evolving a rich notation for the anusaraka output language. It is important to build an intelligent user interface which can assist the user and avoid catastrophe whenever necessary.

The main contribution of anusaraka is to separate language based analysis from world knowledge based analysis. This separation, even though generally implicit in NLP work, has not been recognized explicitly. As a result, most machine translation systems end up being ad hoc. Such ad hoc systems are difficult to build when a large team is involved, and cannot evolve or grow modularly.

More specifically, this thesis discusses and points out several similarities and differences in the syntax of Kannada and Hindi. It shows that the differences can be handled by enriching Hindi with some additional notation essentially resulting in some new function words. Differences in lexical usages in Kannada and Hindi are also handled by introducing additional notation giving wider or narrower meaning to content words.

1.7 Outline of the Thesis

This thesis has five chapters. They can be summarised respectively as: what is anusaraka, what it is not, why it works, why it does not work, and what do we learn from all this. More seriously, the first chapter defines the problem this thesis is trying to solve. As the reader would have noticed, the problem being solved is an engineering problem. Such a problem needs every theory or practice that is useful in solving the problem. The disciplinary boundaries are unimportant.

In Chapter 2 we discuss why the approach followed in the thesis has not been pursued earlier. We look at existing literature in the disciplines of Linguistics, Computer

Science, etc. and their relationship to the problem at hand. We speculate on why modern theoretical research in India has remained alien and aloof from practice.

Chapter 3 contains basic material on anusaraka. The approach is laid out in detail with respect to Indian languages, in general, and Kannada-Hindi, in particular. Some of the problems faced and solutions that worked are described for grammar and lexicon.

In Chapter 4, failures or catastrophes in anusaraka output are discussed . These are cases where the human reader might be misled or fail to understand the target language text. How such cases can be handled and the tradeoff between mild and serious catastrophe is discussed.

Finally, in Chapter 5, there is a presentation on how anusaraka can be used as a measuring instrument by the linguists. We conclude by discussing future directions.

Chapter 2

Pitfalls of the Current Indian Research

This chapter is a kind of literature survey. It discusses existing theoretical results of apparent relevance, but which turn out to be of little practical relevance. Some of the results are of apparent general relevance to NLP (like complexity research), others are of relevance to English (e.g., CFG parsing), and yet others have universal claims (e.g., Chomskyan models), but seem to have artificial fit when applied to Indian languages. None of these results turn out to have much relevance to the problem at hand. (Some have doubtful relevance to even English and general NLP.) This chapter, therefore, goes a step beyond normal literature survey and discusses why the right kind of literature is not getting produced. In other words, what has gone wrong with research. Some of the comments apply to the Indian scenario and we try to identify reasons because of which research in India is so cutoff and irrelevant to practise.

2.1 Glamour

2.1.1 Speculative Research

There is a tendency in research to avoid laborious but necessary work of data collection and analysis. The work of data collection is even looked down upon. In language related areas all this is evident in the almost total lack of good contemporary dictionaries in Indian languages, a lack of corpora, a lack of any exhaustive lists of syntactic phenomena or syntactic differences among various Indian languages, etc. On the other hand, in the name of theoretical work speculative claims are made on very meagre amount of data.

2.1.2 Fashionable Topics

To understand the role of "fashions" in science, we have to look at the dynamics of how research directions are set. It is widely accepted and known that in research, the goals must be set by the individual scientists. Yet there is a need to coordinate and plan the research being conducted in a society. To bridge this gap, there are many subtle and not so subtle mechanisms. Chief among them is "fashion". From time to time different problems and sub-areas within a discipline gain prominence, so to say become fashionable. These are determined by mainly two factors. First, with the advancement of knowledge within the discipline or in the allied disciplines, certain sub-areas appear to hold greater promise for advancement. In other words different problems in those sub-areas seem amenable to solution. Immediately, these sub-areas become fashionable. Second, when practice (such as in industry) faces problems, or a new advancement holds promise for practice, funds become available for solving the problem or related problems; and for work in the concerned sub-area. This leads to a flurry of activity and the sub-area becomes fashionable. Modern times are full of such examples. The second reason is the dominant reason today for setting fashions in science and technology. Fashions translate into special conferences, special issues

of journals, etc. leading to greater publication opportunities for work in the fashionable sub-areas.

The problem is that fashions for science are set today in the West. As a result, they pose those problems that are related to problems being faced by the practice in the west. A result of this is that a large amount of time is spent in our country working on the problems which do not have any relevance to our society.

Almost all the theoretical work in Linguistics in India has followed the fashions in the West. It has been almost solely concerned with universals—laws or properties that hold for all languages of the world. As a result, it has missed on crucial results that hold for Indian languages (Bharati et al. (1992a)). One should remember that typewriter would never have been invented if work had started from the point of a universal machine for printing all languages/scripts.

Most of the theoretical work on syntax in India has adopted models and theories that were designed for English and European languages. They are really misfits for Indian languages. This has been done primarily because of fashion and shows a lack of deep understanding of the issues involved. Some examples are discussed in the next section.

2.1.3 Premature Formalization

Formalization pertains to defining terms and concepts precisely (which usually means, unambiguously and independent of context). Main purpose of formalization is to facilitate communication so that what one researcher is saying can be understood unambiguously and precisely by the others.[1] Modern scholarship has started giving inordinately high importance to formalization. The importance of insight and intuition has been reduced. This is bound to be very harmful to any scientific activity. One should recall that it was insight

[1] Another purpose of formalization is for clarity of thought.

and intuition of Newton and Leibniz that led to the discovery of calculus. It was not formalized until a hundred and fifty years later when the concept of limit was defined. For a hundred and fifty years, it continued to be used as a practical tool by the physicists without formalization!

There is an increasing tendency among our researchers to start developing elaborate formalisms, with little content. Such an activity is unproductive. (Lot of calories but no nutrition!) It can discredit an entire area or approach.

2.2 Some examples of Glamour

2.2.1 Context Free Grammar Parser

In case of Indian languages, primary information (related to what is called the gross meaning) is not in the position of words, but the in case endings or postpositions. For example, the gross meaning shown for (2.1a) remains unchanged in the sentences (2.1b) to (2.1f) in Hindi. In these sentences, the noun groups can come in any order, only their postpositions (ne, ko, etc.) remain unchanged. This simple observation is sufficient to reject the usefulness of CFG for Indian languages.

(2.1a) laDake ne laDakii ko phuula diyaa.
boy -ne girl -ko flower gave
(The boy gave a flower to the girl.)

(2.1b) laDakii ko laDake ne phuula diyaa.

(2.1c) laDakii ko phuula laDake ne diyaa.

(2.1d) phuula laDakii ko laDake ne diyaa.

(2.1e) phuula laDake ne laDakii ko diyaa.

(2.1f) laDake ne phuula laDakii ko diyaa.

Another important aspect of Indian languages is that heads can be identified easily. Therefore, sequence of operations in parsing should be different.

On the other hand, majority of parsers and grammar formalisms are designed assuming English to be a typical language. Most of the effort is directed at handling long distance dependency or movement, which is irrelevant to Indian languages.

If one works with a modern Western "fashionable" formalisms, for parsing Indian languages, one not only misses out and does not take advantage of the features of Indian languages (namely postpositions, recognition of heads, etc.), but one is also likely to pay a price. (See Bharati et al. (1993c) for alternatives.)

2.2.2 Chomskyan Linguistics

The Chomskyan generative enterprise addresses the problem of language acquisition by child. This is not of interest at the current stage of development of NLP and Machine Translation. The argument given below follows Bharati et al. (1992b).

Goal of the generative enterprise is to characterize the initial state of knowledge of language ("abstract innate mechanisms") that allows a human child to acquire grammar for a language because of his or her intimate association with a speech community. Since the child is capable of learning any language depending solely on the association, the innate mechanism is independent of any particular language. The focus of research is on the innate grammar or universal grammar. This research is pursued to the exclusion of research on background knowledge and other cognitive abilities and factors. In fact Chomsky argues forcefully for the autonomy of syntax, and stresses that the object of language study is grammar and not those components which relate to other cognitive aspects, their relationship with language not withstanding.

There are two major consequences of the above position that are important to us

here. They are described in the next few paragraphs.

In the generative enterprise, language is not studied from the point of view of communication; the emphasis is on the notion of grammaticality, i.e., what native speakers consider to be grammatical. As a result, semantics (assignment of meaning to sentences) and pragmatics (purpose of utterance) take a back seat. According to Shieber (1988), generative linguists are only concerned with three aspects of meaning: theta role assignment, quantifiers, and binding and anaphora. Issues relating to word sense disambiguation, structural ambiguity, finding referents of generalized anaphora, relation between modifier-modificand, tense and time, type token distinction, discourse analysis, etc. are ignored. Very little attention is paid to representation of meaning. Where generative linguistics deals with meaning, it is to the extent it impinges on grammaticality.

The emphasis in generative enterprise is not on writing of grammars for any particular language but on the search for universals. Most of the research effort is spent in taking a current set of universals as proposed by Chomsky (such as in Transformational Grammar, Government and Binding) and showing how they work or do not work for various languages. A fairly standard set of language phenomena are taken and the universals posited by the theory are shown to explain them. Dasgupta (1991) argues how several different grammar formalisms can explain a given phenomena, and it is wrong to talk in terms of "the" grammar or universal grammar. It has been argued by Patnaik and his group (See Geetha (1985) and Jain (1990)), that in trying to explain Indian languages by the current set of universals (in GB), several natural notions are not made use of, while unnatural notions (such as "subject") are introduced.

The more important practical consequence of the search for universals has been the neglect of writing of grammars. It has even been argued that the goal of generative enterprise is the study of grammars for languages, not the languages themselves. This is

in contrast to the requirements of computational linguistics, which requires that grammars for particular languages be written.

2.2.3 Complexity Research and NLP

Complexity research has produced widely diverging results for language processing. We will first discuss the results and then the implications.

Complexity of an algorithm or a program specifies the amount of time and memory resources required by it. To characterize complexity independent of the speed of the computer on which the algorithm is running; it is defined in terms of the size of the input. In other words, given an arbitrary input of size n, the complexity gives the time and memory the algorithm takes as a function of n. Most commonly one hears of *worst case complexity* where the complexity is given in terms of the worst possible case for an input of size n. If we consider order of complexity where input n tends to infinity, it is called *worst case asymptotic complexity.*

If there are two algorithms that perform the same task, that is, produce the same output for the same input, then one which has lower complexity is considered better. Of course, the danger is that, usually only the worst case asymptotic complexity analysis results are available; hence they are likely to be used without appreciating the conditions under which they hold. An algorithm that has poorer worst case complexity might have a better average case complexity.

The notion of worst case asymptotic complexity of a program can be generalized to worst case asymptotic complexity of a grammar formalism. We are free to design the best possible parser for the grammar formalism. Complexity of the formalism is the worst case complexity of such a (best possible or best known) parser for an arbitrary grammar and string.

In 1980s, the question whether natural languages are context free assumed importance (became fashionable). Associated with it was the promise that CFG parsers are fast ($O(n^3)$); hence, NLs could be parsed efficiently [2] (See Gazdar et al. (1985)) . Even though this question was answered in the negative (See Pullum (1988)) , the next major claim for restricted grammar formalism came from Joshi (1985) with respect to Tree Adjoining Grammar (TAG). It was claimed that lexicalized TAGs were mildly context sensitive and had just the right locality property to describe natural languages. Relatively efficient parsing algorithms are known for TAGs ($O(n^6)$). LTAGs currently lead in the race for restricted grammar formalisms.

At another location at roughly the same time, work was being done by Tomita in trying to build practical efficient parsers for CFGs. The $O(n^3)$ complexity for CFG parsers was considered as too high to be acceptable. He and other researchers had in mind LR parsers for programming languages which have the complexity of $O(n^2)$ in the worst case and turn out to be linear in practice. This work led to the discovery of Tomita's algorithm (1986) for parsing CFGs. It constructed an LR table with possibly multiple entries in the table. Tomita's algorithm could use such tables and had an efficient representation of packed forests for possible parses. In the worst case (for artificial grammars) the parser would perform worse than $O(n^3)$ but Tomita's parsers for English, in practice operated in linear time. This suggests that English is probably a subset of CFG. Similar results are discussed for free word order languages in Bharati et al. (1990b),(1993a).

To complete the current picture, there are Berwick's results which are completely divergent from the above two. Barton, Berwick and Ristad (1987) showed that NL parsing problem is NP-complete! To state more precisely, if NL has lexical ambiguity and agreement, then the recognition problem, let alone parsing problem, is NP-complete. This means that no matter what grammar formalism is used, one cannot do better than in solving an NP-

[2]This question also has implications for human cognitive capacity for language.

complete problem (for which no polynomial time algorithm is known).

One can see that the complexity results show a wide divergence: from linear in practice to NP complete in theory.

This suggests that something is seriously wrong in using complexity theory to answer questions in practice. The basic model of using worst case analysis seems inappropriate. In case of NLP, what makes it particularly bad, perhaps, is the fact that we are dealing with complexity of grammar formalism, in other words worst case for an input sentence "for the worst case grammar".

To characterize an entire framework as "too complex" based on "worst case" seems incorrect. What is needed is a finer measure. Worst case measure is too coarse to be applicable. Even though it is sometimes mentioned in the preamble that worst case asymptotic complexity might not say something useful about practical implementations, the same edict is forgotten (by the authors as well as the readers). Formalisms are accepted or rejected based on such analyses.

For example it is known that language *wcw* (where w is an arbitrary string consisting of symbols a and b ($w\varepsilon$ {a,b}*)) is not a context free language. It is in a higher class which has a higher complexity (which requires greater amount of processing time or memory). Now the language *wcw* itself may not be complex to parse (in fact, it is not), but such a language and its grammar would be called "inferior" to CFL_s and CFG_s.

Equally seriously, perhaps the basic computational model is inapplicable to NL parsing. What it says, at best, is about grammar formalisms (or classes of languages). In doing so, stack is considered more basic than, say, a queue. This need not hold in an actual parser. For example, *wcw* can be recognised easily with a queue.

In cognitive modelling, considerations of short term memory etc. become very important. Ease of parsing (by humans) might as well be determined by the small size of

short term memory. Thus, Dutch cross-serial dependencies (discussed so much in literature because they are not covered by context free grammars), might actually be easier to parse by humans (Joshi, 1990).

All this suggests that what the complexity theory, is trying to do is like characterizing the flying abilities of aircrafts by the number of wheels they have.

2.3 Infrastructural Difficulties

Besides glamour and glamorous research, there have also been infrastructural difficulties in building anusaraka like system in the past. This might be another reason why such a system was not built earlier.

The first major difficulty pertains to the large amount of language data that needs to be prepared and organized. Any real life system must be fairly comprehensive in its coverage of grammar, lexicon etc. Lexicon size must be at least around 20,000 or so, for the system to be of any use. Testing and tuning of system requires availability of a large corpus. These are not easy tasks. They can be accomplished only by the team work of several individuals and groups.

The second major difficulty is regarding availability of reasonably powerful computers. Until recently, such computers were very expensive and not easily available. Powerful computers are needed not only by the user, but in developing the system in the first place. Development of systems requires easy accessibility and generous amount of computer time. Testing and tuning of lexicon and grammar requires repeated execution of system with large corpora. Fortunately, the rate of development in computer power is such that powerful enough computers are already available. They can be used to develop the anusaraka system. The availability of powerful desktop computers will help deliver anusaraka system to the users.

2.4 Accessibility to Sanskrit Literature—Constructivism and Situatedness

We have already discussed why an anusaraka or anusaraka like system was not built earlier for Indian languages. The first major reason related to glamour and the second to infrastructure of people and computers.

Here, we address the question as to why the traditional Indian scholarship on language has not been used in NLP and modern language studies. There seems to be a natural case for taking a serious look at traditional Indian grammar (Vyakarana) of say Paninian tradition. A comprehensive grammar was written for Sanskrit by Panini, and most Indian languages are very close to Sanskrit. As a result, insights from the Paninian grammar written for Sanskrit should be applicable to modern Indian languages. It also turns out that the goals and concerns of Paninian grammar match very well with those of NLP. Why has it not been looked at? Besides the reasons of glamour discussed earlier, there are reasons of accessibility as well.

The vyakarana literature is not easy to access by a scholar trained in a modern paradigm. The problem can be traced to what is called constructivism in Cognitive Science. A scholar in one paradigm, must understand the concepts belonging to a new paradigm by constructing and deconstructing his earlier paradigm (Mosaic, 1992).

To give one example of the problem of communication between two paradigms consider the concept of karta karaka. If somebody translates it to agent theta role, the most important insight of the Paninian framework is lost. This cannot be brushed aside as a problem of mis-translation. When one looks at concepts in another paradigm, one begins by "translating" or understanding them in terms of the closest concept available in the already known paradigm. Gradually, differences between the approximate translation and the original concept are studied to see whether the differences are material in light of the

questions that are being raised. If indeed the new concept differs from the translated concept materially, a new concept would be introduced (e.g., karta karaka is needed as a new concept in Western Linguistics). This leads to constructing a new model in the reader's mind.

The second problem relates to situatedness. A concrete application at hand serves to provide a vantage point from which to look at a knowledge system. If a problem, particularly a concrete or a practical problem, is to be solved, then looking at a knowledge system from this viewpoint usually yields new insights. Thus, the problem of building a machine translation system or writing a computational grammar, say, for modern Indian languages provides a unique opportunity of looking at Indian Vyakarana and Nyaya on one hand, and Western Linguistics and Logic on the other.

Chapter 3

Anusaraka: a Detailed Study

3.1 Introduction

Natural language is used for communication between a writer and a reader. Strings (or utterances) of natural language contain information that the writer (or the speaker) intends to convey to the reader (or hearer). The reader analyses the string and is able to retrieve the information coded by the writer. This is an everyday occurrence which needs no argument; usually, both the processes of generation and analysis of the string are carried out with relative ease.

A language string explicitly codes only a part of the information that the writer wants to express. It follows therefore, that the remaining information is not coded, and therefore, not contained in the string. (It has to be inferred or deduced from the background knowledge or context, etc.) The principle of information flow makes a clear distinction about what is and what is not, in the input string. It says that one cannot extract information that is not present.

Take for example the sentences (3.1) and (3.2) in Hindi. The preferred readings are shown. They use the preference rule that karta is to the left of karma. Actually, sentences (3.1) and (3.2) are ambiguous between who does the eating: the cat or the fish. The

ambiguity arises because both nouns have the same feminine gender (marked as 'f.') as the verb, and agreement does not help us in distinguishing one noun from another. The information about who does the eating is simply not coded unambiguously in the string.

(3.1) billii machalii khaatii hei.
cat(f.) fish(f.) eats(f.)
(Cat eats the fish.) (preferred reading)

(3.2) machalii billii khaatii hei.
(The fish eats the cat.) (preferred reading)

As a further example, see sentence (3.3) where the gender marking makes it clear that the cat eats the chicken and not vice versa. Karta being on the left is overridden.

(3.3) murgaa billii khaatii hei.
cock(m.) cat(f.) eats(f.)
(Cat eats the chicken.)

Even though in a story it might be plausible for the chicken to eat a cat; no amount of background knowledge can bring about this interpretation in sentence (3.3). This is simply a result of the fact that what is explicitly coded in the string cannot be overridden by background knowledge.[1]

To make the point more stark, note that sentence (3.4) states unambiguously that the 'chicken does the eating.'

Selectional restrictions which are sometimes used in NLP systems to disambiguate these sentences, are really a part of the background knowledge. This can be seen from

[1]If the explicit coding conflicts with the background knowledge, it is the latter which must be given up, or alternatively, look for errors in transmission of the string, errors in coding, etc.

the fact that selectional restrictions can easily be overridden by constructing a different background. For example, the selectional restriction that chickens do not eat cats, can be overridden, say in a story, where there is a ferocious chicken going around eating cats.

(3.4) murgaa billii khaataa hei
(m.) (f.) (m.)
(The chicken eats the cat.)

Sources of information in any particular case may be quite diverse; one must carefully analyse the various assumptions involved in the extraction of information before trying to generalise them for arbitrary situation. In the above examples, who does the eating is coded in gender number person agreement. (The problem arose when there was ambiguity, i.e., there was agreement of verb with more than one noun.) The coding is also done through postposition markers. In (3.4a) 'ne' marker shows that billii (cat) did the eating:

(3.4a) billii ne machalii khaayii.
cat(f.) ne fish(f.) ate(f.).
(Cat ate the fish.)

We would like to draw a clear distinction between two processes: information extraction process and inferential process. The former process extracts information explicitly coded in the sentence, while the latter makes use of background knowledge. This distinction is important while dealing with the machine, because it is very difficult for the machine to deal with background knowledge. Note that no claim is being made about normal human cognitive processing. In all likelihood, the two processes go together each helping the other. However, there are instances of not so "normal" processing (e.g., reading a rule book or law book, reading in an area where background is insufficient) where the two processes may not go together very closely.

The following example shows the role of background knowledge in drawing conclusions, that is, interpreting the sentence. This requires making choices among the several interpretations that are possible based on information coded in the string. Here again the pieces of knowledge might be quite diverse.

(3.5) mohana ki shaadii usakii saasa kii laDakii se huii.
[maamii/buaa]
mohana's marriage his mother-in-law 's daughter-with took place
[mother's brother's wife/father's sister]

(mohana's marriage took place with his mother-in-law's daughter.)
[mother's brother's wife/
father's sister]

Under normal circumstances the Kannada person will immediately interpret it to mean

mohana ki shaadii usakii buaa kii laDakii se huii.
father's sister's daughter

In this case the "saasa"(mother-in-law) reading has been ruled out because it will violate the usual norm of conversation. The sentence is almost a tautology. Culturally, both readings: maamii(mother's brother's wife) and buaa(father's sister) are acceptable as they are permitted by custom. However, the 'maamii' reading is not accepted because of Gricean maxim of quantity (Levinson, 1983). If the speaker really meant maamii's daughter then he could have referred to her as maamaa's (mother's brother's) daughter; why take a circuitous path to indicate the relation when a shorter path is available?[2]

[2]However, note that if 'maamaa' (mother's brother) is no more, or maamii has remarried or a contrast was being shown between say maami versus mausi then use of maamii over maamaa might be preferred. Thus, the implicatures establish preferences only.

The above reasoning implicitly assumes that the speaker is co-operating with the listener, but in general this need not be so. One can also easily imagine a situation where the speaker does not want to answer the listener's query and simply replies with (3.5) meaning 'saasa' as mother-in-law.

The following sentence in Kannada codes the information that the leaf is related to eating. It does not say what the relationship is.[3]

(3.6) K: mohana tiMda eVleVyannu eVseV.
mohana eaten leaf throw
(Throw the leaf on which Mohana had eaten.)
H: mohana ne khaayaa_hei_jo_usa patte ko pheMka do.
mohana ne eaten which leaf throw
(Throw the leaf on which Mohana had eaten.)

Any Kannada person will immediately get the following meaning of the above sentence in a proper context.

H: mohana ne jisa patte para khaayaa hei usa patte ko pheMka do.
(Throw the leaf on which Mohana had eaten.)

On the other hand a person who is not aware of the fact that in Karnataka, food is quite frequently served on banana leaves may find the sentence a bit puzzling. So in this case the part of the information determining the relation between the leaf and the act of eating comes from the knowledge of the social custom. We will have more to say on this later.

The above example shows that there are large number of ambiguities in language use. A sentence might represent codings of many alternative pieces of information. There

[3]Note that in the roman notation for Kannada, 'eV' and 'oV' stands for the short 'e' and 'o' vowels respectively as shown in the orthographic notation chart at the beginning of the Thesis.

might even be orderings on (or preferences among) the alternatives, based on language knowledge. For examples, economy or maxim of quantity might place preferences over the alternatives as in the case of 'maamii'. However, these can be overridden.

Thus, in reality the situation is quite complex. The explicitly coded information by the speaker appears only as one of many alternatives obtained from the language string.

3.2 Nature of Language Knowledge

As mentioned earlier, there are two broad kinds of knowledge that are needed by a processor (a human reader or any other processor) to interpret or get the meaning of an input language string. They can be further broken up as:

1. Language Knowledge
   - **a.** Grammar
   - **b.** Lexicon
   - **c.** Pragmatics and discourse

   etc.
2. Background knowledge or world knowledge
   - **a.** General world knowledge (including common sense knowledge)
   - **b.** Domain specific knowledge (includes the specialized knowledge of the area about which communication is taking place).
   - **c.** Context (verbal or non-verbal situation in which communication is taking place).
   - **d.** Cultural knowledge

All this knowledge helps in extracting information out of a language string. As discussed earlier, language knowledge is used in extracting information explicitly coded in the string.

Background knowledge helps in reducing ambiguity by ruling out certain readings and thereby facilitates extraction.

Different processors have different properties regarding processing. Humans normally find it easy to use world knowledge. Language knowledge is less easy, particularly for the second language user. He might not have enough knowledge of grammar or lexicon. In fact, learning the lexicon takes a large amount of time. When a user does not have some part of knowledge, it affects the amount of work that needs to be done to extract information. This has to do with redundancy of coding; when a simpler method of extracting information is not available a longer and more complex method has to be used. Also many ambiguities have to be carried forward while processing, making the task of processing harder.

Computers are just the reverse: they are very poor at handling world knowledge and much better at handling grammar and lexicon. Thus, given a language string, computers would find it easiest to extract information using lexicon and grammar.

There seem to be at least two levels of coding complexity in language—both grammar and lexicon. The first level is that of simple complexity which is easy to process. In fact, it requires less time and memory for processing (and is also easy to learn). The second level (and other levels, if necessary) is that of more complex coding. Both codes appear on the surface form, namely the sentence; former level may be called *outer* surface coding and the latter level as *inner* surface coding.

Let us look at some examples of the outer and inner surface codings in the following sections.

3.2.1 Grammar

Some of the coding of information is quite explicit in grammar. For example, the postposition marker 'ne' in Hindi specifies the karta relation between a verb and a noun:

(3.7) laDake ne laDakii ko kitaaba dii.

boy erg. girl-to book gave

(Boy gave the book to girl)

Thus, occurrence of 'ne' specifies that the word before it in the sentence is the karta of the sentence (which is the agent in this example). This coding is simple and the extraction by a processor is immediate.

On the other hand, certain information is coded deeply in grammar. It is not readily available without a lot of processing. An example of this is the '-kara' (having-done) construction for verbs in Hindi:

(3.8) laDakaa khaanaa khaakara ghara gayaa.

boy food having-eaten home went

(The boy went home after having eaten food.)

Here, the verb khaa (eat) does not have an explicit karta in the sentence. 'laDakaa' (boy) is the explicit karta of 'gayaa' (go). The karaka sharing rule states that karta karaka of khaa (eat) is the same as the karta of the verb 'gayaa' (go) which is modified by khaa in this sentence. By this reasoning, 'laDakaa' (boy) is also the karta of 'khaa' (eat).

The above rule, though it works without exception, is not part of the outer coding. This rule does not seem to be known very directly to either the speakers or the linguists. But that alone is not the main reason. Application of this rule requires first identifying which verb is modified by the verb with '-kara' suffix and then accessing its karta. This in turn requires greater amount of processing. Moreover, if the information about modified verb and karta is not coded on the surface, there maybe no way to reach a unique answer.

3.2.2 Lexicon

The lexicon too has outer and inner coding. Each word has a nuclear meaning such as:

```
H: kuuda  (to jump)
H: uDa    (to fly)
K: neVgeV (to jump)
K: haaru  (to fly)
```

However 'koti haaritu' (literally, monkey flew) is used in Kannada, for monkey's jumping. Similar uses and conventions get established though reasons may be varied. Thus, besides the nuclear sense, a word has a "sense space". This sense space specifies: what other senses does a word have, in what ways the word can be used, in what contexts it is permissible, in what contexts it is inappropriate, etc.

The nuclear sense is an outer coding, while the sense spaces are part of the inner coding. Learning the former is much easier compared to the latter. The latter clearly requires much greater amount of memory per word. Also it requires greater amount of processing time; because a larger amount of memory has to be searched and context has to be matched to determine the sense.

The above observations are testified by second language learners. They learn the primary or the dominant sense of a word much easily than they learn the nuances and other ways that the word can be used.

3.2.3 Background Knowledge

Background knowledge is not coded in the language string, however, it is useful in extracting information from a string. Consider the example sentence (3.9). It is ambiguous in the sense who peeled and ate the bạnana.

(3.9)

raama ne haatha se Ciilakara kelaa khaate hue baMdara ko dekhaa.

Rama erg. hands inst. having-peeled banana eating monkey acc. saw.

(Rama saw a monkey eating a banana having peeled it with his hand.)

The karaka relations explicitly marked in the sentence (i.e., in outer surface coding) are (Bharati et al. (1993a)) for details.):

dekha (see)

karta: raama

karma: baMdara (monkey)

Ciila (peel)

karana: haatha (hand)

khaa (eat)

karma: kelaa (banana)

We have already seen that there are well defined rules in the language to determine the karakas not marked in the outer surface coding. They are part of inner surface coding. Some of the other karakas can be determined depending on which verb modifies which other nouns and verbs. There are four different possibilities. The four possibilities permitted by language rules are given below in which peel and eat modify, respectively:

(a) eat, see (i.e., peel modifies eat, and eat modifies see)

(b) eat, monkey (i.e., peel modifies eat, and eat modifies monkey)

(c) see, see (i.e., peel modifies see, and eat modifies see)

(d) see, monkey (i.e., peel modifies see, and eat modifies monkey)

The karaka sharing rule mentioned earlier does not tell us who is doing the peeling uniquely, though it might narrow down the answer to a smaller number of choices. This is where the world knowledge comes to the rescue. If we already know from background knowledge that the monkey was eating the banana (corresponds to 'eat modifies monkey'), we might readily infer that the monkey probably peeled it too. A whole chain of reasoning from grammar has been cutoff and a much shorter reasoning has been used.

The above has the danger, however of making the reader miss some other interpretations. Here for example, if we know that monkey was eating the banana, it corresponds to cases (b) and (d) above. Thus the language coding with the above background knowledge allows us to conclude (b) or (d) corresponding to peeling being done by monkey or Rama, respectively. The second interpretation (d) was missed when we assumed that in a boy and monkey situation the peeler is also the eater.

3.3 Anusaraka and Information Flow

We have discussed in Chapter 1 how the respective strengths of the computer and the user can be combined to overcome the language barrier. Computer is good at dealing with lexicon and grammar, while the humans are good with background knowledge but find it difficult to deal with lexicon and grammar of a new language. Therefore, the language knowledge load must be taken up by the computer, while the background knowledge load is to be taken up by the user.

We are now ready to pose the problem of anusaraka in terms of information flow. Suppose the machine has *no* background knowledge, it has only the language knowledge of the source and target languages. From the source language text it must extract the information coded in the text. This information is presented in a language close to the target language. The reader by making use of his background knowledge should be able to

get the "intended" meaning in the original source text.

The output language is close to the target language, but has a syntax of its own. It may even be called a dialect of the target language. The reader will usually require some learning of the dialect. But this learning time will be negligible when compared to the learning time of the source language.

The main problem of anusaraka is how to present the information extracted from the source text in the target language. As it does not use any background knowledge, it may have to take an incomplete "picture" obtained from the source and recode it in the target language. Here is an example to illustrate the problem. Sentence (3.10) in Telugu expresses 'we will go' without specifying gender, and 'we' is inclusive of hearer or addressee.

(3.10) Telugu: manamu veVLataamu.

We (inclusive) will go.

A similar construction in Hindi requires that gender be coded but inclusion of hearer is not coded:

H: hama jaayeMge/jaayeMgii.

we will go(m.)/go(f.).

There is a problem, therefore, in coding exactly the same information from one language to another. This problem arises because we want to generate a sentence of about equal length and paralleling the sentence construction wherever possible. Clearly, it may be possible to express the same information by a longer prose, but if the size is much bigger than the original, it makes it difficult for the reader to get the same meaning as in original source language. Flavour of the original is also lost.

The anusaraka answer is to deviate from the target language in a systematic manner. First, new notation is invented and incorporated. Thus, we can decide to have hama+

(we+) to mean inclusion of hearer. Second, we may relax some of the conditions in the target language. For example, we might give up agreement in our "dialect" of the target language:

```
@H: hama+ jaayeMge.
    we+   will go.
```

Alternatively, we can choose an existing word in Hindi with somewhat similar meaning except in anusaraka Hindi it will always have the sense of inclusive we as in:

```
@H: apana jaayeMge
```

Some of the constructions of the source language may also get introduced in the target language. Existing words in the target language may be given wider or narrower meaning(as in "apana"). Thus, the new language will require some learning; but the output will contain the same information which was coded in the source sentence. Still, catastrophes can occur either due to a lot of inner surface coding or due to interactions among codings, as discussed in the next chapter. On-line help can provide access to the detailed analysis of the source text, if the reader desires.

As it is important to present the target text with exact and the same amount of information coded in the source text, there are two important parameters of relevance:

1. Size of the output text.

2. Richness of output notation

It has been observed that the size of the output should be roughly comparable to the size of the input for easy comprehensibility. For example, if for every ambiguous word or phrase, a separate target language word or phrase is produced (separated by slashes, say) size of the

output goes up sharply and comprehensibility goes down. Here, the notation is minimal, and the size is large. If on the other hand, a suitable notation is introduced, it will become possible to express the different ambiguities more compactly. However, the learning time for the notation will increase. In the limiting case, the output notation can be the source language itself. In such a case, the size of the output is equal to the size of the input. Learning time, however, is as great as learning the source language itself.

Thus, there is a tradeoff between the size of the anusaraka output and the richness of notation in the output (over and above the target language). Making an appropriate choice is an important design issue. In the remaining chapter, we will take the case study of Kannada-Hindi anusaraka and look at some problems and their solutions.

For understanding the case study, it is important to distinguish between content words and functional words. Grammar for Kannada and Hindi (and other Indian languages) is primarily defined in terms of the functional words. Role played by position of words is only secondary. Content words on the other hand, refer to objects in the world (real or hypothetical world), and are discussed under the lexicon.

3.4 Problems of Grammar—A Profile of Kannada-Hindi Case study

We have taken a brief look at the kinds of problems that arise in trying to present information in the target language. Here we will look in detail at a case study—about how to present in Hindi the information extracted from a Kannada text.

3.4.1 Giving up Agreement in Anusaraka Output

Hindi has an agreement rule which can be stated as follows:

Gender, number and person (gnp) of the verb agrees with the gnp of the karta if it

has ϕ vibhakti[4]; otherwise the gnp of the verb agrees with the karma if it has ϕ vibhakti; otherwise the verb takes masculine, singular, third person form.

When the input Kannada sentence has an ambiguity in identifying karta (or karma whichever is relevant for agreement) and the gnp of the two candidate kartas (or karmas) is different, an ambiguity will appear in the gnp of the verb.

Similarly possessive modifiers of nouns need information about gnp of the related nouns. There are a number of cases where it is not easy to compute. Sometimes, the relevant information may not even be available.[5]

Kannada has three genders (masculine, feminine and neuter), whereas Hindi has only two (masculine and feminine). Consequently, certain amount of information loss is bound to occur in going from Kannada to Hindi. To avoid this loss, it will be necessary to provide additional notation to mark the neuter gender. Here is an example to give its significance:

In Hindi, for karma-kartru prayoga, typically separate verb forms are available; but in Kannada, frequently a single verb stem is employed. Thus, on the surface, it seems that it does not distinguish between "daravaajaa khulaa" and "daravaajaa kholaa". However, in practice one can distinguish between the two usages by paying attention to the gender marking on the verb; in the first case it will be neuter gender while in the second, typically, it will be a non-neuter gender.[6]

```
K: baagilu teVreVyitu.
```

[4]Karta having ϕ vibhakti means that it is not followed by a postposition marker (a function word).

[5]There is a temptation to provide correct agreement according to standard Hindi grammar, wherever it is possible to do so without much effort and not bother about agreement when it is not possible to do so easily; but such a policy will be potentially confusing to the user. From the utility point of view, it always helps to keep the working of the system simple and to provide the user a faithful picture of the working of the system.

[6]"daravaajaa kholaa" (opened the door.) is an incomplete sentence in isolation but it can occur in discourse in response to the question "raama ne kyaa kiyaa?" (What did Rama do?)

@H: daravaajaa khulaa[kholaa].

(The door opened.)

K: baagilu teVreVdanu.

@H: daravaajaa khulaa[kholaa].

((He) opened the door.)

Loss of agreement is not expected to be too jarring to Hindi speakers in view of their exposure to various varieties of non-native Hindi heard everyday as propagated by television, radio and films.

3.4.2 Language Bridges

Apart from agreement there are only three major syntactic differences between Hindi and Kannada. Surprisingly all of these can be taken care of by enriching Hindi with a few additional functional particles or suffixes as shown below. Thus they can be viewed as lexical gaps or functional word gaps.

3.4.2.1 "ki" construction

An embedded sentence in Hindi, which is a complement of a verb, is put after the main verb and the complementizer 'ki' precedes immediately to the complement unlike in Kannada. For example, consider the following sentences: [7]

(3.12a) H: raama ne kahaa ki mohana kala aayegaa.

Rama said that mohana tomorrow come-fut.

(Rama said that mohana will come tomorrow.)

[7] K, H, and E specify the language of the sentence as Kannada, Hindi and English, respectively; '!H' stands for gloss in Hindi: '!E' for gloss in English, usually understood and not shown in the Thesis, K@H for Kannada anusaar Hindi, that is, output produced by the Kannada-Hindi anusaraka; @H as an abbreviation for K@H when the source and target languages are clear form context; etc.

K: *raama heLidanu eneVMdareV naanu maneVgeV hogutteneV.

!H: raam kahaa ki meiM ghara ko jaauuMgaa.

(Rama said that he will go home.)

Note, that the above Kannada sentence sounds odd to a Kannada person, because of the order of the constituents. In Kannada the complementizer 'eVMdu' occurs after the complement— exactly a mirror image of what is found in Hindi. Therefore, the Hindi complementizer 'ki' cannot be used to replace Kannada complementizer 'eVMdu'. However by using "eisaa" instead of "ki" we can easily avoid this problem, as illustrated below:

(3.12b) K: mohana naaLeV baruvanu eVMdu raama heLidanu.

!H: mohana kala aayegaa eisaa raama ne kahaa.

Mohana tomorrow come-fut. that Rama said.

'eisaa' construction is a proper construction in Hindi; only it is used less frequently. In the dialect of Hindi produced by anusaraka, however, this will be the normal construction used.

3.4.2.2 "jo" construction

Kannada has a large number of adjectival participial phrases or clauses which convey information about tense, aspect etc. but they do not have information about karaka relations. In sentences (3.13) and (3.14), Kannada codes information about tense on the verbs eat and make.

(3.13) K: raama tiMda camacavannu toLe.

©H: raam khaayaa thaa cammaca dho Daalo.

(3.14) K: raama tayaarisida camacavannu toLe.

H: raama banaayaa thaa cammaca dho Daalo.

As mentioned already, these sentences do not contain information about the relation eat has to spoon or make has to spoon. It is the background knowledge that provides the relation between them. For example, the general world knowledge may be used to say that spoon is the instrument of eat (and is not eaten) in (3.13). In the next sentence (3.14) spoon could be the instrument or theme of make.

Hindi, on the other hand, has only two participial phrases viz. yaa_huaa and taa_huaa which code perfective and continuous aspects only, e.g.,

(3.15) H: khaayaa huaa phala

eaten fruit

(3.16) H: khaataa huaa laDakaa

eating boy

Thus, anusaraka would be able to use these constructions in Hindi only when the tense information in Kannada is appropriate. But what about the other tense, aspect and modality informations? There is a syntactic hole in Hindi!.

There is another problem, however. The two participial phrases in Hindi have coding for karaka relations which is absent in Kannada. yaa_huaa codes karma[8], while taa_huaa codes karta. Participial phrase (3.15) indicates the fruit that was eaten, while (3.16) indicates the boy who is eating. Thus, Hindi is poorer than Kannada in coding tense, aspect, modality information, while richer in coding karaka information in case of adjectival participles. But this compounds the problem for anusaraka. Using these constructions

[8]More correctly, yaa_huaa codes karma in case of sakarmaka or transitive verbs, and karta in case of intransitive verbs. But sometimes a clause containing a sakarmaka verb with its karma behaves like a akarmaka verb as in 'kaama siikhaa huaa naukara'.

in Hindi would mean putting in something that is not contained in the source language sentence.

The answer lies in identifying another construction in Hindi and creating a correspondence between it and the constructions under consideration in Kannada.

Hindi has a relative clause—the 'jo' construction which allows both tense and karaka information to be specified. For example, to say 'wash the spoon with which Rama has eaten' we can write any of the following sentences (3.17a), (3.17b) or (3.17c):

(3.17a) H: raama ne jisa cammaca se khaayaa thaa usko dho Daalo.
raama erg. which spoon inst. eaten that wash
(Wash the spoon with which Rama has eaten)

(3.17b) H: raama ne khaayaa thaa jisa cammaca se use dho Daalo.

(3.17c) H: raama ne khaayaa thaa jisase usa cammaca ko dho Daalo.

To express the same information as in the Kannada sentences (3.13) and (3.14), we can invent a notation on the lines of the jo-construction as follows:

(3.17') raama ne khaayaa thaa jo_* vaha cammaca dho Daalo.

The vibhakti markers (i.e., the functional words se, ko etc.) are replaced by '*'. 'jo_*' could even be replaced by 'so' to produce a kind of colloquial Hindi in some region.

(3.17") raama ne khaayaa thaa so cammaca dho Daalo.

Unlike the first case, this idea takes some time and effort for the Hindi reader to get used to.

3.4.2.3 "ne" construction

This "ne" construction is a peculiarity of only the Western belt languages in India. In case of the present or past perfective aspect of the main verb in Hindi sentence, "ne" is used with the karta:

(3.19) H: raama ne phala khaayaa.
Rama erg. fruit eat-past
(Rama ate the fruit.)

In case of 0_gayaa TAM label, 'ne' is not used.

(3.19') H: raama phala khaa gayaa.

Therefore, we can postulate a new word TAM "yaa'" with same semantics as "yaa", but which does not use "ne" construction; with this TAM we can express the corresponding Kannada sentence more faithfully as: raama phala khaayaa'.[9] Other constructions are given in Appendix-A.

3.5 Structure of Anusaraka

Before discussing other problems with grammar, we will first look at the various components of anusaraka system to get a feel for the various kinds of tasks to be performed.

Structure of the anusaraka is shown in Fig. 3.1. A source language sentence is first processed by morphological analyzer (MORPH). The MORPH considers a word at a time, and for each word it checks whether the word is in the dictionary of indeclinable words. If found, it returns its grammatical features. It also uses the word paradigms to see whether

[9]It may be of interest to note that the "yaa" pratyaya in Hindi corresponds to "kta" pratyaya in Panini's grammar and so the new proposed pratyaya "yaa'" will be natural counterpart of the "ktavatu" pratyaya in the Sanskrita grammar.

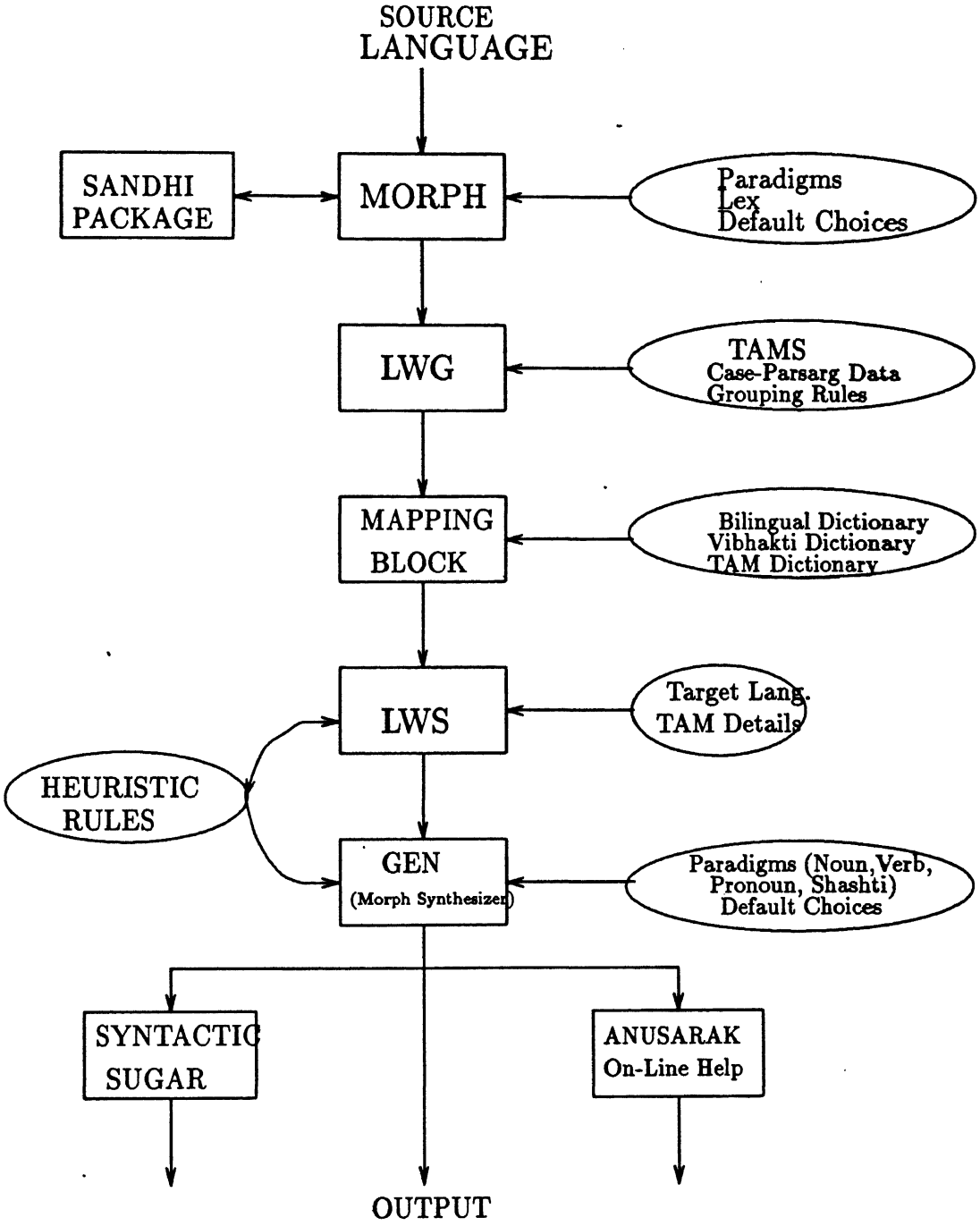


Figure 3.1: Block Schematic of Anusaraka

the input word can be derived from a root and its paradigm. If the derivation is possible, it returns the grammatical features associated with the word form (obtained from the root and the paradigm). In case, the input word cannot be derived, it is possibly a compound word and is given to the sandhi package to split it into two or more words, which are then again analysed by MORPH.

The output of MORPH is given as input to the local word grouper. Its main task is to group functional words with the content words based on local information such as postposition markers that follow a noun, or auxiliary verbs following a main verb. This grouping (of root/stem plus inflections), identifies vibhakti of nouns and verbs. The vibhakti of verbs is also called as TAM (tense-aspect-modality) label.

After the above stage, sentential analysis can be done. Current anusaraka does not do this analysis because it requires a large amount of linguistic data to be prepared. Also, since the Indian languages are close, 80–20 rule applies to vibhakti. Use of vibhakti produces 80% "correct" output with only 20% effort.[10] Sentential parser can be incorporated when large lexical databases are ready.

The next stage of processing is that of the mapping block. This stage uses a noun vibhakti dictionary (see Appendix-C), a TAM dictionary which is made with the help of Kannada TAM chart (see Appendix F), and a bilingual dictionary. For each word group, the system finds a suitable root and vibhakti in the target language. Thus, it generates a local word group in the target language.

The local word groups in the target language are passed on to a local word splitter (LWS) followed by a morphological synthesizer (GEN). LWS splits the local word groups into elements consisting of root and features. Finally, GEN generates the words from a root and the associated grammatical features.

[10]It is our estimate that the sentential parser will improve the performance only marginally for going from South Indian languages to Hindi.

3.5.1 User Interface

Anusaraka output is usually not the target language, but close to it. Thus, the Kannada-Hindi anusaraka produces a dialect of Hindi, that does not have agreement etc. It can be called a sort of Dakshini (southern) Hindi. Some additional notation may also be used in the output. Certain amount of training is needed for a user to get used to the anusaraka output language.

The role of the anusaraka interface (or on-line help in Fig. 3.2) is to facilitate the reading of the output by a reader. It should keep track of what concepts have been introduced to the user, and also provide on-line help when the user faces a problem.

Depending on the nature of interface, namely, an ordinary user interface, or intelligent user interface, one can have HAMT (human aided machine translation) and/or machine translation (MT). (See Fig. 3.2.)

3.6 Problems of Grammar Continued

3.6.1 Local Word Grouping

Indian Languages have relatively free word order; still there are units which occur in fixed order (See Bharati et al. (1990c)). The most important examples of these are the main verb followed by auxiliary verb sequences and nouns followed by postpositions. We term such units as verb groups and noun groups respectively. It may be noted that verb groups and noun groups will be sub parts of what are called verb phrases and noun phrases respectively. However in our formulation we do not use the concepts of noun phrases and verb phrases because

1. the concept of verb phrase does not seem to be natural for Indian languages and

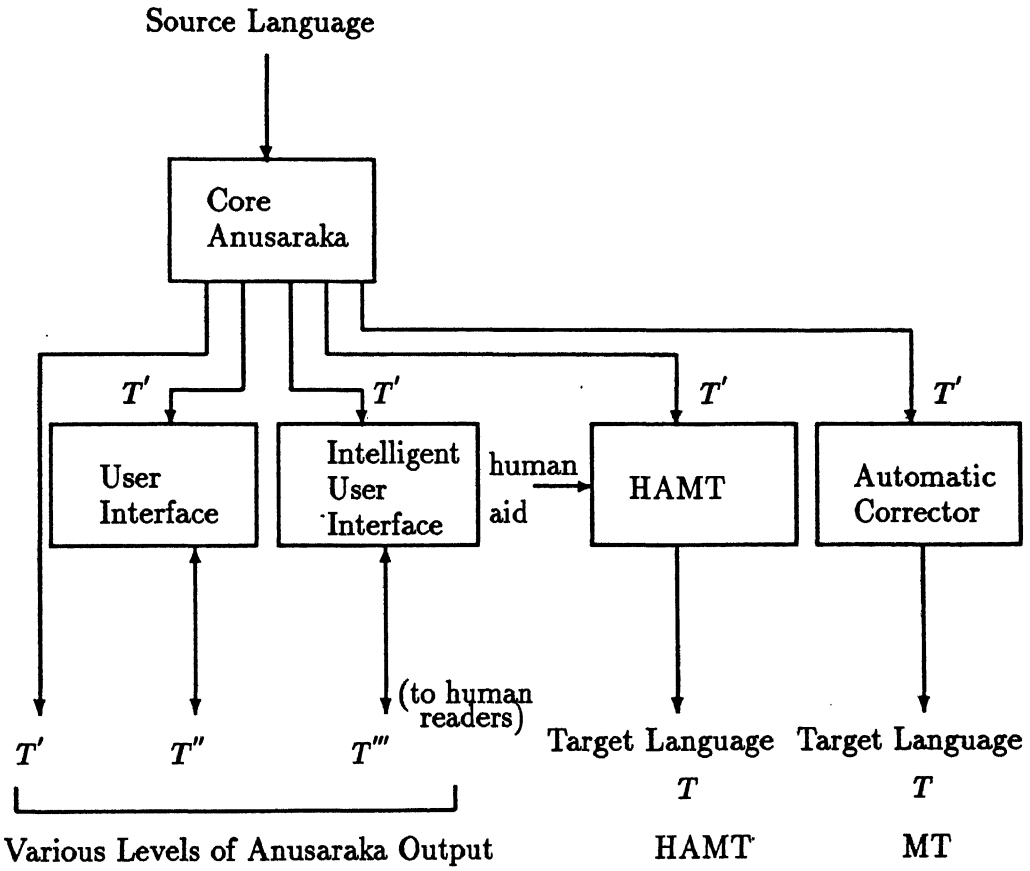


Figure 3.2: Different Interfaces for Anusaraka

2. from the computational point of view recognition of noun phrases and verb phrases is neither simple nor efficient.

On the other hand noun groups and verb groups can be formed using only local/surface information and more importantly they provide sufficient information (viz. 'prayoga' and 'vibhakti transformation rules') for further processing of the sentence according to Paninian karaka theory. So the local word grouping provides all the necessary information with minimum computational effort.

Another important point about our local verb grouping is that we do not attempt to distinguish all the fine shades of semantics associated with these verb sequences. Our experience with Hindi and Kannada data suggests that to a large extent these fine shades are conveyed by identical conventions among Indian languages and so for anusarana or translation purpose we need not disambiguate them. This approach is also consistent with Indian grammatical analysis where meaning is extracted in several layers with increasing precision.

Groups once formed may not be split later, hence the groups are formed out of adjacent words, that are known to be unambiguous. The verb grouping is done based on the possible verb sequences that may occur and information about agreement. The noun groups are composed based on the form of the noun and the following postposition.

As described in the last section, there are typically two kinds of word groups consisting of nouns and postpositions, and main verbs and auxiliary verbs, respectively. The exact groups formed, vary from language to language. For example, in Hindi, sequence of verbs, main and auxiliaries, occur as separate words and hence have to be grouped together. In Kannada, they mostly occur together, conjoined into a single word: as a result, the analysis is left to morphology rather than to the local word grouper.

However, there are instances where Kannada too requires local word grouping (LWG). The following examples show the necessity of LWG for Kannada.

- hattu halavu (ten many):

  In Kannada 'hattu' means ten and 'halavu' means many.

```
K: raamanu eneno  kaaraNagaLannu heLi,        siitaLannu hattu halavu
   Rama   various reasons-acc.  having said, Sita-acc.  ten   many
   sala   noDidanu.
   times  saw
   (Rama saw Sita many times  by giving  various reasons.)
@H: raama na_jaane_kyaa_kyaa kaaraNoM ko kaha kara, siitaa
    ko dasa[caDha] kaii baara dekhaa.
```

  These two words be grouped and translated as 'aneka' or 'dasiyoM' (literally, many or tens) as shown below.

```
@H: raama na_jaane_kyaa_kyaa kaaraNoM ko kaha kara,
    siitaa ko aneka baara dekhaa.
```

- ceVlli hoguttittu (having spilled gone):

  Consider the following sentence:

```
K: naanu nanna manassannu jaatreVya kaDeVgeV harigoVTTiddareV
   I     my    mind-acc.  festival  towards  flow-to
   koVDadalliya niiru ceVlli hoguttittu.
   pot's         water spill  go-past_perf.
  (If I had allowed my mind to flow towards festival,
   the pot's water would have spilled.)
```

@H: meiM meraa mana ko melaa kaa tarapha bahanaa se ghaDaa
meM kaa paanii chalaka[chitaraa] kara jaataa_[thaa].

However, if we group the two words together then it is possible to translate them as spilled ('chalaka jaataa').

@H: meiM meraa mana ko melaa kaa tarapha bahanaa se ghaDaa
meM kaa paanii chalaka[chitaraa] jaataa.

Note that since both Kannada and Hindi use verb sequences to show tense aspect and modality (where the auxiliary verb is usually an independent verb), even without LWG we are getting reasonable output. If the output is produced without LWG, the output would be totally unacceptable.

- kuDiyuttaa baMdanu (drinking came): Again, consider a similar sentence.

K: madhyeV madhyeV alli taanu irisidda doVDDa taMbigeVyiMda
middle middle there himself kept big vessel-from
niirannu baggisikoVMDu kuDiyuttaa baMdanu.
water-acc. having tilted drinking came
E: (In between he kept on drinking water by tilting the big vessel
which he had kept there.)
@H: madhya madhya vahaaz aapa[svayaM] rakhaa huaa baDaa loTaa se
paanii ko jhukaa kara piitaa huaa aayaa.

After local word grouping, and substituting 'piitaa gayaa' (kept drinking) instead of 'piitaa hua aayaa' (come drinking), we get

@H: madhya madhya vahaaz svayaM rakhaa huaa baDaa loTaa se paanii
ko jhukaa kara piitaa gayaa.

The above rule is valid provided the main verb is 'not a physical-motion' verb as in the above sentence. However, this rule does not hold good for the 'physical motoin' verbs like 'douDanaa' (to run), 'calanaa' (to walk), 'reMganaa' (to crawl), and 'sarakanaa' (to move). The following example illustrates this.

```
K: raama shaalegeV naDeVyuttaa   baMdanu.
   Rama  school-to having walked came
   (Rama came to school by walking.)
Without:  raama shaalaa ko' calataa_huaa aayaa.
With   :  *raama shaalaa ko'  calataa gayaa.
```

- idde iruttadeV:

```
Without: raha_kara_hii_rahataa_hei
With   : rahataa_hii_rahataa_hei

K: naayi maneVyalli idde            iruttadeV.
   Dog   house-in   be-perf.-emph. be-pres.-3pn
   (Dog will be definitely in the house.)

Without: kuttaa ghara meM raha_kara_hii rahataa hei.
With   : kuttaa ghara meM rahataa_hii_rahataa_hei.
```

As one can see from the above examples, LWG helps in the following ways:

1. Minor syntactic differences between the pair of languages are taken care of. For a syntactic construction in the source language (which can be handled by LWG) a single word group is obtained. This can be replaced by an appropriate word group in the target language. This improves the readability of the target text. For example, minor

syntactic differences when taken care of, improves the ease of reading as illustrated by the following examples:

a. kaa + kaaraNa → ke_liye

K: raamana saluvaagi siita yaava kaSTavannu anubhavisaluu sixXa.
Rama's for Sita whichever trouble-acc. suffer-to ready
(Sita is ready to suffer whichever trouble for the sake of Rama.)
Without: raama kaa kaaraNa siitaa kisa kaSTa ko bhoganaa_* siddha.
With : raama ke_liye siitaa kisa kaSTa ko bhoganaa_* siddha.

a. kaa → 0

K: raama maneVya tanaka hogi matteV hiMdirugidanu.
Rama home's up to having gone again returned
(Rama returned after going up to home.)
Without: raama ghara kaa taka jaa kara phira vaapasa_aayaa.
With : raama ghara taka jaa kara phira vaapasa_aayaa.

2. Consider the following usages found in Kannada as they are handled. For example the Kannada construction:

K: raamana haageV
!H: raama_kaa usa_prakaara
Rama's like that

translated by anusaraka as

@H: raama ke jeise

Similarly take the Kannada phrase:

K: hattu halavu

!H: dasa kaii

ten many

For this phrase, anusaraka can produce

(i) H: dasiyoM

E: tens

or

(ii) H: aneka

E: many

The former preserves the flavour while the latter gives the straight meaning.

3. Certain word level ambiguities are removed by LWG. This happens when local information around a word rules out some other meanings. For example:

K: tiMDi tiirtha

H: naaStaa tiirtha

breakfast pure water

With local word grouping, anusaraka produces the correct meaning:

@H: naaStaa paanii

break fast

A table of word sequences that form local word groups is given in Appendix-D.

3.7 Problems of Dictionary

In this section, we will discuss our approach towards dictionary making for anusaraka.

3.7.1 Evolutionary Operation

For building a dictionary for anusaraka, we have followed the evolutionary approach. It consists of the following three steps carried out sequentially: start from existing material, make incremental changes, and provide for feedback.

To start making the dictionary, we take material that is already available. For example, to build a Kannada-Hindi dictionary for anusaraka, we begin with a Kannada-Hindi bilingual dictionary meant for human beings.

The next step is to decide the kind of changes that need to be made in the dictionary. For example, going through the various senses of words listed in the dictionary one should group them appropriately.

Having decided on what changes to make, one *should not* try to build a perfect system in a single step. Many things can be postponed to future. Improvements can be incremental and can be made from time to time.

For such an incremental building, feedback is crucial. The feedback helps in judging whether the improvements made to the dictionary, lead to a better anusaraka system. It also helps in suggesting what new kinds of changes can be made.

One way to ensure proper and realistic feedback is through a useful working system. Thus, one should try to keep a real working anusaraka system at every stage. The actual feedback can then be used in improving the dictionary. A working system is also necessary to provide something useful to society and seek society's support for the programme.

Finally, good engineering requires that the most time consuming jobs be identified

and done early so that commissioning is not delayed.

3.7.2 Issues Involved

The following issues must be kept in mind while building or improving the dictionary. Examples are from Kannada to Hindi.

3.7.2.1 Substitutability

A bilingual dictionary designed for human beings gives the meaning(s) of a source language word in terms of the target language. The emphasis is on conveying the meaning. The human reader on learning the meaning(s) can make sense out of a given source text. Similarly, examples of usages of the word might be given in the dictionary to facilitate the learning process.

Requirements on the bilingual dictionary for anusaraka are different. The emphasis is on *substitutability*. For every word in the source language, the anusaraka dictionary should give a word in the target language, whenever available, such that on replacing the source words in a text by their equivalents in the target language, the information is conveyed to the reader. For example, 'mari' in Kannada is used to refer to a baby or young of an animal or a bird. A dictionary meant for humans shows the entry as:

```
mari::noun:: pashu   pakSiyoM_kaa baccaa
             animal  bird's       baby
```

If such a dictionary is used by anusaraka for replacing words, this is what happens:

```
 K: beVkku  mari heVttitu.
@H: billii ne  pashu pakSiyoM kaa baccaa janma_diyaa.
    cat    erg. animal bird's      baby   gave-birth
```

(The cat gave birth to a baby of animal or bird.)

The problem is that in the above context, it is understood that a cat will give birth to only a kitten. There is no separate word for kitten in Hindi. The fact that a long phrase is used, Grice's maxim suggests that it must be significant. Therefore, the reader is led to think that the cat must have given birth to something other than a kitten! This is an example of how different phenomena, lexical and pragmatic, interact in complex ways to produce unexpected results.

The anusaraka answer is to look for a substitutable word. The anusaraka dictionary, therefore, simply contains:

```
mari::noun::1.baccaa{pashu_pakSiyoM_kaa}
```

Because anusaraka tries to preserve information, it uses curly brackets to show what information is present in the source language word.

Information in square or curly brackets may become useful in another context. Consider the following sentence occurring in a fairy tale where a queen actually gave birth to an animal baby:

```
K:  raani  mari   heVttaLu
@H: raanii baccaa{pashu_pakSiyoM_kaa} janma diyaa
    queen  baby  {animal bird's     } birth give(past)
    (The queen gave birth to a baby{of animal or bird}.
```

Here it might be necessary to take the other interpretation.

In a nutshell, the anusaraka dictionary tries to find a word in the target language that is substitutable. If none is available, it uses a word that can be substituted in usual

contexts, with additional notation to preserve the information (for unusual contexts). Thus, it tries to preserve the information together with substitutability.

3.7.2.2 Minimizing Sense Clashes

For each source language word in the dictionary, one should try to cover its different senses by as few entries of the target language as possible. This helps control the size of the anusaraka output. Because one way to present the output is through disjunction of different entries; as the number of entries increases, the size of the output also goes up. Two example entries are shown below, where each number represents a different sense and '/'shows alternative words for a sense.

If we look at the dictionary meanings of the homophonous word 'bhaava', we find mainly two unrelated senses, i.e., bhaava (manner of being) and bahanoii (elder sister's husband). The homophonous nature of this word is because, in Kannada there already exists a word 'bhaava' (elder sister's husband) and had borrowed later the Sanskrit word 'bhaava' (manner of being, etc.) which have same phonetic content but have unrelated senses. Such potential sense clashes can be resolved by making two entries (or more if necessary). The modified entry for the word 'bhaava' is as shown below:

```
bhaava1::noun::1.bhaava/vicaara/abhipraaya 2.manovikaara/svabhaava 3.astitva
              1.manner of being/idea/opinion 2.mental state       3.existance

bhaava2::noun::1.bahanoii/bahana_kaa_pati
                1.elder sister's husband
```

3.7.2.3 Much Wider Meanings

A target language word with a much wider meaning coverage than the source language word can lead to catastrophes. A reader of the anusaraka output may be led astray or will be left wondering as to what was meant. Additional notation should be used to indicate restrictions on meanings.

3.7.2.4 Right Selection

In case there are many entries for a word, a selection of entries may become necessary. What this means is that in the output produced by anusaraka, only some of the entries will be shown with a suitable notation. The other entries will be accessible to the reader but only on demand. On-line help will be available for this purpose, and can be exercised by the reader if he so desires (in case of difficulty, etc.).

Statistical information about usage of the word in various senses can be determined from corpora. This information can be used to arrange the senses in a particular order and to perform selection.

3.7.3 Nuclear Sense

A question we might ask, is what is it that we are capturing by doing the above. In the answer, we explore nuclear sense and sense spaces.

Usually, a word has several meanings, but frequently they are all close to each other. If we imagine that each sense is a point in the sense space (or conceptual space), then the senses of a word would frequently be all clustered together. This cluster is called the sense space spanned by the word, or simply its *sense space*. The *nuclear sense* of a word is a sense that, in a way, explains or gives rise to all the senses of the word. One can also define it as

the sense that is equidistant from all other senses of a given word. It can be said to be at the "centre" of the sense space of the given word.[11] For example, the Kannada verb 'seru' has a large number of senses covered by many different words in Hindi as shown below. However they are all related: to come near, to meet, to enter, etc. The Hindi word 'juda' (to join) is the nuclear sense of 'seru'.

```
seru::verb::1.+juda/pahuMca/nikata_aa/mila_jaa/samaa/aa/pravesha_kara
                to reach/to enter/to be included
              2.pasaMda_ho/acchaa_laga/bhaa/Thiika_jazca/caaha/pyaara_kara
                to agree with(as food)/to like
              3.ikaTThaa_ho/peiMTha
                to join/to meet
```

Nuclear sense is important because it is a conceptually unifying device for the various shades of meaning one sees in a word. In anusaraka, the nuclear sense can help in two ways:

1. While making a dictionary for anusaraka one should try to identify the nuclear sense of a word. The nuclear sense can be used to obtain a suitable "substitutable" word in the target language. (If we look for words for each of the senses, it may be difficult to arrive at a single word in the target language.)

2. The users of anusaraka can learn to extract information from the output more effectively, if they understand the idea of the nuclear sense and how it is used in anusaraka dictionary.

It is appropriate to mention here that nuclear sense or a concept close to it, is an important part of lexicon knowledge. It is one of the first things a second language learner

[11] We will not attempt to define nuclear sense any further here.

learns. Knowledge of the sense space of a word is more voluminous and complex, and involves all kinds of nuances and idiosyncratic usages. Certain usages are permitted but other close usages are not, for example 'uDaa' (fly) in Hindi:

H: paani kii buuMdeM uDii.
water drops flew
(Drops of water flew.)

H: *baMdara uDaa.
*monkey flew.
(Monkey jumped)

H: mohana ke sira ke baala uDa gaye

Finally, some words may have two or more nuclear senses. This can be observed when the plotting of their senses gives more than one cluster of senses. This could typically happen because of the following reasons:

1. Historical: The senses were close in History but because of change in practices they now seem far apart. For example, Kannada 'oleV' means palm leaf, letter or ear ornament. It reflects the historical practice of using palm leaf to write letters and as material out of which ear ornaments were made (Acharya, (1991)).

   Similarly, 'tiMgaLu' means the month or the moon for obvious reasons. 'kaalu' means leg or quarter. This is so because one leg out of four among four legged animals is one fourth or quarter of the total.

2. Borrowing of foreign words: When a word is borrowed from another language that is spoken or written in the same way as an existing word, the meanings may be unrelated.

For example, 'hari' in Kannada means God Vishnu or is a verb (to flow or to tear). The noun sense corresponds to the word borrowed from sanskrit, where as the verb sense corresponds to an original Kannada word.

3. Obsolete letter merging: Sometimes changes in script or pronunciation lead to the same word. For example, in Kannada the word 'hari' has two different nuclear verb senses: One for intransitive verb meaning to flow, walk, run, crawl etc. and another for transitive verb meaning to tear, break, etc. These have arisen out of two different words one spelled as 'hari' another spelled differently. They became the same when an obsolete letter in the script was merged with another.

3.7.4 Some Further Tips

Here are some additional tips for adapting a dictionary.

1. Etymological: Etymological information can be included in the dictionary to aid the user when he seeks assistance using on-line help. We have already seen some examples in the last section under Historical.

2. Statistical: Statistical information can be included in the dictionary to aid in sense selection. These have been discussed earlier. As a first step, obsolete senses and rare senses may be marked.

3. Domain: Dictionary can also be tailored to a domain. This tailored dictionary can be used whenever necessary. Such a tailoring will eliminate (or reduce priority among) senses which do not occur (or occur less frequently) in the domain. Conversely, certain (new) senses of words may become more dominant in a domain. These will have to be given special consideration.

Use: Usages of words in different senses should be listed in dictionary much like in a learner's dictionary. These would be useful in providing on-line help to the user.

Cultural: Certain words with strong cultural significance can have special explanation provided with them. Again this would be useful in on-line help. Examples are

a. haldii kuMkuma: In South India, it is a custom to invite married woman whose husband is alive (sadhavaa) to the house and to offer 'haldii kuMkuma' along with fruits and flower on all auspicious days. Unless, the reader knows this cultural details, he might find it difficult to understand the output.

K: niivu iMdu namma maneVgeV arishina kuMkumakkeV banni.
@H: aapa aaja hamaaraa ghara ko' haldii kuMkuma ko' aayiye.

b. chuaachuuta or shuddha: In South India, elderly people from orthodox family (particularly from Brahmin community) after taking bath, will observe a state called 'maDi' for worshipping or praying God. If anyone touches a person who is in 'maDi' state, then he/she has to take bath again and attain 'maDi' state again to continue the prayer. Unless, the reader knows this custom, the sentence given below looks odd.

K: siitaa, amma iiga maDiyalli iddaaLeV. maguvannu avaLa hattira biDabeDa.
adu enaadaruu ammanannu muTTidareV amma matteV snaana maadabekaaguttadeV.
@H: siitaa, maaz aba shuddha meM hei. baccaa ko usakii{strii.}
paasa mata_choDa. vaha' jo_bhii_ho maaz ko chuunaa_se
maaz phira[matavaalii_strii] snaana karanaa paDegaa

Exact equivalence: When a general word is used (e.g., baccaa) restrictions along with it are listed in square brackets. The two together give an exact equivalence, as discussed earler.

7. Adding the missing present usage:

```
Before: vyavasaaya::noun:: vyavasaaya/udyama/rojagaara/prayatna
                           profession/business/trade/endeavour

After: vyavasaaya::noun:: 1.+kRshi 2.vyavasaaya/udyama/rojagaara/prayatna
                          1.agriculture
```

8. Making multiple entries: If causative verbs derived from two different verb roots exhibit identical form (homophony), then make two separate entries as aLisu1 and aLisu2 in the dictionary. The derived form 'aLisu' (←'aLi+isu' to ruin and 'aLu+isu' to make to cry) is the causative verb of two different verb roots, viz., 'aLi' (to be ruined) and 'aLu' (to cry).

```
Before: aLisu::verb::1.miTaa/naSTa_kara/maara_Daala 2.rulaa
                     1.to ruin/to destroy/to kill   2.to make to cry
 After: aLisu1::verb::1.miTaa/naSTa_kara/maara_Daala
        aLisu2::verb::1.rulaa
```

9. Retaining the original word whenever possible: If a word exists in both source language and target language and is having the same nuclear sense, then retain the original word in target language also. For example, the dictionary entry for Kannada word 'kaala' is given as 'samaya'. Since in Hindi also the word 'kaala' is there with same meaning, 'samaya' is changed to 'kaala' in the dictionary. However, take the word 'anumaana' which is there in both Kannada and Hindi. In Kannada 'anumaana' means doubt and in Hindi it means inference or estimate. It is obvious that the above tip does not apply here.

3.8 Summary

In this chapter, we have had a detailed look at the working of anusaraka. It required a clean separation between language knowledge and world knowledge and use of the concept of information flow. We looked at the problems of grammar and lexicon. Through a case study of Kannada-Hindi anusaraka, we have discussed language bridges, local word grouping, nuclear senses, etc.

Chapter 4

Catastrophes

4.1 Introduction

In the previous chapter we saw that if we adopt the criterion of intelligibility as the measure of the syntactic difference, then Kannada syntactic structure is not much different from Hindi syntactic structure and the apparent mismatches can be easily bridged by simple devices of local word grouping and addition of some functional words and units which closely resemble the existing ones. To a large extent this suffices to produce an output from a Kannada text which can be easily understood by a Hindi speaker after a brief learning or adaptation phase. But occasionally the reader will come across sentences which he will not be able to decipher. Not only this, sometimes he may even misunderstand some of the sentences. In this chapter we shall examine methods to avoid such catastrophes.

4.2 Catastrophe

As indicated above, catastrophes in anusaraka output are said to occur when the reader either

1. fails to comprehend the meaning of a sentence, or

2. misinterprets the meaning of a sentence.

The former is called a *mild catastrophe* while the latter is called a *serious catastrophe.*

The following is an example of a mild catastrophe:

(4.1) K: naagarakoyilannu serida taruvaayu eVllaruu kaaphi tiirisideVvu.
©H: naagarakoyila_ko juDaa huaa anaMtara saba bhii kaaphii khatma_kiyaa.
Nagarkoil-acc. reaching after all coffee finished
(After reaching Nagarkoil, all (of us) finished coffee.)
H: naagarakoyila pahuMcane ke baada hama saba ne kaaphii khatma kii.

In the sentence given below, serious catastrophe takes place and the user misinterprets the sentence:

(4.2) K: kaanpura nanageV hoVsa uuru.
Kanpur me-to new attach/town
(Kanpur is a new town to me.)
H: kaanapura mere liye nayaa shahara hei.
©H: kaanapura mujhe nayaa lagaa/jukaa[gaazva].

A reader is likely to interpret this as:

kaanapura mujhe nayaa lagaa.
(Kanpur seemed a new town to me.)

The correct interpretation is

kaanapura mere liye nayaa gAzva.
(Kanpur is a new town to me.)

Let us look at some more examples of serious catastrophes which have been seen to occur in Kannada-Hindi anusaraka:

(4.3)

K: nanna hattirada saMbaMdhi oVbbaru pratidina namma toTakkeV baruttiddaru.

@H: meraa nahiiM_lagaa[caDhaa]_huaa saMbaMdhii eka pratidina hamaaraa baaga ko' aataa_[thaa]

The problem here is that Kannada word 'hattirada' can have two different meanings: 'paasa_kaa' (close) and 'nahiiM lagaa[caDhaa] huaa' (not attached). Only the latter is listed in the dictionary. The point in this example is that the original Kannada sentence does not lead to any catastrophe to a Kannada speaker even if he does not know the former meaning, while in case of the anusaraka reader it leads to a serious catastrophe. The reason is that an anusaraka reader is willing to tolerate a lot more. A Kannada speaker (if he does not know the first meaning) would find the sentence construction odd. He will recognize that he does not understand the sentence because of probably an unknown meaning, but no serious catastrophe will occur. The anusaraka reader, on the other hand, is unfamiliar with the conventions of Kannada. As a result, he is willing to tolerate and give interpretations (misinterpretations, in this case) thus leading to serious catastrophe.

Serious catastrophe sometimes occur because of an ambiguity which is not present in the source language, but introduced in the target language by anusaraka. For example, 'lagaa' in Hindi has two different derivations.

1. lagaa (lagaa + imperative) — to plant. For example:

(4.4) kheta meM phasala lagaa.
field in crop plant
(Plant the crop in the field.)

2. lagaa (laga + yaa) — felt. For example:

(4.5) mujhe lagaa ki kheta meM phasala hei.
I felt that field in crop is
(I felt that the crop is in the field.)

'lagaa' in anusaraka output may get misinterpreted, particularly because the second sense ('felt') seems to be applicable more often. Note however that in the original Kannada sentence, the laga-yaa sense is not present. That appears only because lagaa in Hindi is ambiguous. This reasoning arises particularly, if the source language also has the tendency to drop "I" in sentence construction with 'felt' and drop "is" in stative or copula sentences.

We have found that 'lagaa' is extremely prone to serious catastrophe because in the sense of 'feel' it can go with virtually any sentence. Responsible, in part, of course is the tendency on part of the reader to quickly give some meaning to the output. This tendency is quite natural, because that is the strategy needed most of the time.

4.2.1 Issues in Catastrophe

This brings us to an issue that seems to be important in catastrophe: If there is an ambiguity in a part of the anusaraka output (because of target language construction, lexical ambiguity, etc.) then it can lead to serious catastrophe when another ambiguity interacts with it. It might be argued that natural language is full of ambiguity. Any text produced by a speaker or writer should have exactly the same problem for a human reader. Why is it that only the anusaraka output has serious catastrophes?

In trying to answer this question, we should observe two things. First, since we are talking of the human reader, the background knowledge can be assumed to be the same. Therefore, the answer does not lie in appealing to differences in background knowledge or

lack of background knowledge. Second, it is not the case that there are never any serious catastrophes when reading is done by readers who are well versed with language. We all know that serious catastrophes do occur, though rarely, in normal language use.

There seem to be two major reasons why interaction of ambiguity leads to serious catastrophes in anusaraka output:

1. The reader of the anusaraka output might not be very proficient. He is a learner of a new language: the target language as produced by anusaraka. For example, the reader of the Kannada-Hindi anusaraka has to learn Kannada anusaar Hindi, that is, a dialect of Hindi that follows Kannada. Learning a language has two major parts: the outer-surface part dealing with grammar and nuclear sense of a word, and an inner part pertaining to deep rules of grammar and sense spaces of words. The latter takes some time to learn. It includes conventions and usages of words. In the absence of the latter knowledge, the normal anusaraka reader is more prone to serious catastrophes. (This is not very different from serious catastrophes that a normal learner of a second language makes. It also suggests that as the anusaraka reader becomes more proficient, he is less likely to make such errors.)

2. Catastrophe avoidance requires deliberate planning. Partly why the readers of normal output (produced by humans) do not have catastrophes is because the writer *recognizes when certain constructions or words might lead to serious catastrophes, and avoids them.* Thus, the reason that there are (almost) no serious catastrophes in normal language use is because the writer has *deliberately* planned for their avoidance. Note that the writer cannot and does not try hard to avoid mild catastrophes. Further research is needed on these issues, if such sophisticated catastrophe avoidance strategies are to be used by anusaraka.

Another important issue relates to the length of string. Consider the following two

sentences discussed earlier which have exactly equal information:

```
K: beVkku      mari                          heVttitu.
@H: billii ne  pashu  pakSii kaa baccaa janma diyaa.
    cat   erg. animal or bird's baby    gave birth
    (The cat gave birth to a baby of animal or birds.)
```

Yet the second sentence leads to a mild catastrophe. This catastrophe can be avoided as discussed earlier. But the issue is why are sentences interpreted differently when the grammatical and lexical information is the same. Answer seems to lie in pragmatics. In the anusaraka Hindi, four words are used 'pashu pakSii kaa baccaa' (baby of animal or bird) for a concept for which original Kannada has one word. Gricean maxim of quantity suggests that since additional words are used, which are not needed normally, there must be something special about it. Hence, the implicature is that perhaps the cat did not give birth to a kitten but some other animal.

4.3 Catastrophe Avoidance Strategies

Here, we will discuss some strategies that can be used to avoid or rather reduce both mild and serious catastrophe.

1. Using notation to restrict meaning. One of the most important methods to reduce catastrophe is to use special notation to convey exact information. When the target language does not have an equivalent word or construction for the source language word or construction, special notation can be invented. Anusaraka would use the notation when needed. This, however, increases the complexity of learning of the output language. Some notation such as ko' or dative marker in anusaraka Hindi are examples of this (see Chapter 3).

2. Displaying danger signals. Serious catastrophe is to be avoided at all costs. One way is to warn the reader whenever words or constructions that are prone to serious catastrophe are encountered. An example could be 'laganaa' (to feel). Whenever, 'laganaa' occurs in the output, a danger mark (such as '!!') will also be shown to blink along with. This would suggest to the reader that he should take care and read carefully.

   A shortcoming of this approach is that the careless reader might ignore the danger mark and not put in the effort required to go over the output carefully. As a consequence, he might misinterpret the sentence.

3. Teach about comparative knowledge of source and target language. Another way to avoid catastrophe is to raise the knowledge level of the reader. He can be taught about the comparative syntax and lexical properties in the source and target languages. As a result, he will be able to recognize structures produced in the output that arise because of input language syntax and lexicon. An example of this would be the ambiguity between o'clock and hour in Kannada word 'gaMTeV'.

   In Kannada, for both o'clock (baje) and hour (ghaMTaa), the word 'gaMTeV' is used. Vibhakti marker helps to disambiguate between the above two senses to some extent as in the following example:

```
K: raama 4 gaMTeVgeV    keVlasa maaDidanu.
   Rama  4 o'clock-dat. work    did
   (Rama did the work at 4 o'clock.)
H: raama ne kaama 4 baje kiyaa.

K: raama 4 gaMTeV     keVlasa maaDidanu.
   Rama  4 hour-nom.  work    did
```

(Rama did the work for 4 hours.)

H: raama ne 4 ghaMTe kaama kiyaa.

The first sentence has dative case marker and the second is in nominative. Now, consider the following sentence which is ambiguous and is giving two readings in Kannada. However, if anusaraka reader is trained about the fact that Kannada word 'gaMTeV' has two senses, the ambiguity of Kannada sentence is reflected in anusaraka Hindi also.

K: raama 4 gaMTeVtanaka keVlasa maaDidanu.
Rama 4 o'clock/hour-up to work did
(Rama did the work up to 4 o'clock/hour.)
H: raama ne 4 baje/ghaMTe taka kaama kiyaa.
@H: raama 4 ghaMTaa taka kaama kiyaa.

Similarly, anusaraka readers must learn the optional usage of 'is' (hei) in Kannada.

K: raama oVbba oVLLeVya huDuga.
Rama a good boy
(Rama is a good boy.)
H: raama eka acchaa laDakaa [hei].

This again requires a larger learning effort on part of the reader.

4. Developing intelligent interface. There is yet another way to avoid or reduce catastrophe, namely, by providing an intelligent interface that tailors itself to the text and to the user. Given a text, it displays its output with special notation whenever necessary. In this, it would make use of the particular kind of text. Some of the information about the text would be inferred by it automatically and some would be obtained by

it from the user. For example, in Kannada the phrase 'paTTaabhiSeka aayitu', is typically abbreviated as 'paTTa aayitu'. However, such a substitution cannot be made generally because otherwise it will lead to serious catastrophe. The system should however be aware of this use of 'paTTa' and be able to provide it to the user when necessary.

```
K: arjunana moVmmaganu-aada    parikSittigeV paTTa      aayitu.
   Arjuna's grandson (who was) Parikshit-to  coronation happened
   (Parikshit who was Arjuna's grandson was coronated.)
CH: arjuna kaa potaa_[huaa] parikSitta_ko' nirdhaara_kaa[paTTa] huaa.
 H: arjuna ke pote parikSitta kaa paTTaabhiSeka huaa.
```

Note that here what compounds the problem is that there are two alternatives 'nirdhaara_kaa (of decision) and 'paTTa' (coronation). The first one is implausible because 'kaa' marker ('of') usually does not come before the final verb. An intelligent interface can keep track of interaction of two problems which otherwise in conjuction may lead to mild catastrophe. However, intelligent interface can help the user by providing more information about such words as 'paTTa'.

The intelligent interface must also have a model of what the reader knows and what he does not know. This model should preferably be acquired through normal interaction. This model can be used to tailor the output to the reader. There is a discussion on it later in the Chapter. This approach requires more work in AI on user modelling etc.

5. Designing to avoid catastrophe. One way to reduce catastrophe is to identify potential problem areas for detailed study and to come up with specific (creative) solutions for each of the areas. This can be done by anusaraka developers by extensive study of outputs produced from corpora.

A design criterion is to maintain one to one mapping of vocabulory (content words and functional units) between source and target texts. To achieve this, one may have to select somewhat less frequently used words or may even have to borrow words from non-standard dialects of the target language. An example is the functional word 'so' which is used in the language bridge for "jo" construction. It has been borrowed from a non-standard dialect of Hindi. This has earlier been discussed under language bridges in Chapter 3.

```
K:  raama haakida   maavina giDa  sattitu.
    Rama  put-ppl.  mango's plant died
    (Rama's planting of mango sapling died.)
@H: raama lagaayaa_hei_jauna_so aama kaa poudhaa maraa.
 H: raama kaa lagaayaa huaa aama kaa poudhaa murjhaa gayaa.
```

4.4 Causes of Mild Catastrophe

As discussed earlier, in case of mild catastrophe the reader has difficulty in understanding the output. Usually, this occurs because of the many possibilities in the output. For example, if three of the content words in a sentence have two meanings each, it generates eight possible sentences. Alternatively, only one sentence is generated with notation (say square brackets and slashes) for possibilities for each of the three words. Consider the Kannada word 'kaayuvudu'.

```
 K: kaayuvudu
@H: taapegaa[intajaara_karegaa/rakSaa_karegaa/
    taapanaa/intajaara_karanaa/rakSaa_karanaa]
```

The verb 'kaayu' has three entries in the dictionary, i.e., as intrasitive verb 'taapanaa' (to heat), as transitive verb1 'rakSaa_karanaa' (to protect) and transitive verb2 'intajaara_karanaa' (to wait). The morph analyses the above word 'kaayuvudu' as 'kaayu + future' and 'kaayu + verbal_noun'. Therefore, we have six alternatives available for the word 'kaayuvudu'.

K: alliya tanaka kaayuvudu ekeV eVMdu aStaralli mRgaalayavannu
there's up to waiting why that meanwhile zoo-acc.
noDi baralu hodeVvu.
having seen to come went
((We) went to see the zoo, thinking why to wait till that time.)

©H: vahaaz_kaa taka taapegaa[intajaara_karegaa/rakSaa_karegaa/
taapanaa/intajaara_karanaa/rakSaa_karanaa] kyoM eisaa usa_samaya
mRgaalaya_ko dekha_kara[xekhe] aane_ko' gayaa.

H: utanii dera taka intajaara kyoM karanaa? yaha socakara cidiyaaghara
dekhakara aane ke liye gaye.

The reader will have a problem in selecting the correct sense out of six alternatives, as is evident from the above sentence. Also the size of output increases and readability decreases.

Consider another example given below. Here the Kannada transitive verb 'oVgeV' has two equal probable senses, i.e., 'dhonaa' (to wash) and 'pheMkanaa' (to throw). This catastrophe can be avoided by making two entries for the verb 'oVgeV' in the dictionary, so that both the senses will come in the output. However, such increase in the number of entries will also increase the size of the output and hence possibility of mild catastrophe in general.

K: avanu kaDDipeVTTigeVyannu nanna meleV oVgeVdanu.
he match box-acc. my on threw

(He threw the matchbox on me.)

GH: vaha'{pu.} maacisa ko meraa uupara dhoyaa.

H: usa ne maacisa mere uupara pheMkii.

When mild catastrophe occurs, the reader has to put in an extra effort to understand the output. He is not misled, however, as in serious catastrophe.

Now we list the causes of mild catastrophe and possible solutions to avoid them.

1. Large number of possibilities. This is the usual reason for mild catastrophe as discussed above. This problem can be overcome by the use of lakshan charts to disambiguate word meanings (See Bharati (1993c)). These charts help to select an appropriate sense of a word. One disadvantage is that they are not always reliable, and sometimes wrong meanings may get selected. The danger is that a few of the mild catastrophes might get converted to serious catastrophe. Thus, there is a tradeoff between having a large number of mild catastrophes or a much smaller number of mild catastrophes but with a few serious catastrophe.

   The development of comprehensive lakshan charts requires a major effort. Such an effort is best done after an initial system is operational so that experimentation with lakshan chart may be done in real life situation.

2. Unfamiliar notation. Sometimes a reader might not know a special notation that appears in the output. The solution is to use suggestive and uniform notation which the user can guess (or not forget, in the first place). The second thing is to produce on-line help.

3. Unknown words. Sometimes the reader may not know a word in the target language that appears in output. The immediate solution is to provide an on-line dictionary for the target language.

This also suggests that anusaraka should be customizable to the reader. For a "simple" user, the anusaraka can use a much smaller vocabulary compared to the normal user.

4. Unfamiliar sentence structure. The readers who are unfamiliar with the structure of the source language will need training. To make this task easier, there is a need to prepare graded output and practice sessions so that learning can be effective and efficient. Examples of unfamiliar sentence structures are given below:

   The first example shows what the anusaraka Hindi reader might have to get familiar with "he too and his book too".

```
 K: avanu aayitu   avana pustaka aayitu.
    he    be-past his   book    be-past
    (He too, and his book too.)
@H: vaha'{pu.} huaa usakaa{pu.} pustaka huaa.
```

   Similarly, consider the following examples where many words are used in their idiomatic usages. Each one involves learning of an unfamiliar structure or usage by the reader.

```
 K: polisanannu noDi          kaLLa   kaalu kittanu.
    police-acc. having seen thief   leg   pluck-past
    (Having seen the police the thief ran away.)
@H: pulisa ko dekha  kara cora peira ukhaaDaa.
 H: pulisa ko dekha  kara cora bhaagaa.

 K: nannannu uttara bhaaratakkeV hoVttu  haakidaru.
    me        north  India-to   carried put-past
    ((They) transferred me to North India.)
```

©H: mujhe uttara bhaarata ko‘ Dho_kara Daalaa.

H: mujhe uttara bhaarata ko bhejaa gayaa.

K: duMdu maaDi tanna svattannu eVlla mugisuva hoVttigeV
Excessively spent his property-acc. all exhausted time-to
xesha eVlla doVDDa kSaamakkeV tuttaayitu.
country whole great famine-to targeted
(By the time all his property got exhausted by excessive
spending, the country was stricken by a big famine.)

©H: ativyaya karake aapakaa[svayaM_kaa] jaayadaada ko saba
samaapta_karanaa_vaalaa samaya ko‘ desha saba baDaa
akaala ko‘ shikaara_banaa.

H: usane ativyaya karake svayaM kii saba jaayadaada ko samaapta
kiyaa hii thaa ki puuraa desha bhayaMkara akaala kaa shikaara banaa.

Here is an example of a very long Kannada sentence that was actually encountered in real text. We have included it here to show the kind of difficulty that sometimes arises simply because of length and interaction of several anusaraka structures.

K: aadareV aakeVyuu raajana manadanneVyaMteV vartisuttiddaruu,
But She-too king's lover-like to-behave
avaLa aMtaraMgada rahasya priitigeV paatranaada
her mind-inside secret love-to role-of
aramaneVya ashvapaalanu idda kaaraNa, raajanaMteVye raaNiyuu
palace-of Groom was reason, king like too queen-too
yocisi, aatane (ashvapaala) bahu kaala badukirali eVMdu
having thought, himself Groom long time live-to-be said

aa phalavannu aatanigeV koVTTaLu aMteV.
that fruit-acc. to him gave said

(But, she (the queen) too was pretending to be king's love, gave away the fruit to a horse-groom at the palace because he had become the candidate of her secret love inside of her heart and like the king the queen too thought he (horse-groom) should live long.)

©H: lekina vaha{strii} bhii raajaa kaa priyaa ke_jeise vyavahaara_karataa_to_bhii, usakii{strii.} aMtaraMga kaa rahasya priiti ko' paatra_[huaa] raajamahala kaa ashvapaala rahaa huaa kaaraNa, raajaa ke_jeise hii raanii bhii soca kara[soce], vaha{pu.} (ashvapaala) hii bahuta samaya[paava_kaa/paazva_kaa] jii le kaha_kara[eisaa/kisa_dina] vaha_phala ko use{pu.}' diyaa eisaa_kahate_heiM.

H: lekina vaha bhii raajaa kii priya hone kaa bahaanaa kara rahii thii, usake gopaniiya prema kaa paatra raajamahala kaa ashvapaala thaa, jisa prakaara raajaa ne raanii kii diirghaayu kii kaamanaa kara ke phala use diyaa usii vicaara se raanii ne ashvapaala ko phala diyaa eisaa kahate heiM.

4.5 User Interface

User interface is one of the most important part of anusaraka. Its purpose is to tailor the output to the user and present it in a suitable form, respond to user queries, provide assistance when needed, etc. The following are the specifications of a user interface.

4.5.1 Specifications

The user interface must try to satisfy the following requirements.

1. Present the output in a suitable form. The display should be such that the target language together with notation are suitably presented. Alternatives should be shown without cluttering up the screen or confusing the reader.

2. Use syntactic sugar as controlled by the user. Syntactic sugar brings the output closer to the target language. For example, if the anusaraka output does not have gender agreement, the syntactic sugar might try to provide agreement whenever possible. Syntactic sugar, in general, reduces mild catastrophes but might introduce serious catastrophe. Syntactic sugar needs to be user controllable—"add according to taste."

3. Set environment. The user interface should be such that it controls the output according to the environment. The environment is determined by the text—its type and nature, and by the user. For example, if the text is a formal report, it requires the use of a particular kind of conventions and vocabulary, and if it is a literary text another kind will be presented to the user. Similarly, if the user has a small vocabulary, the interface would present the output with a smaller vocabulary.

4. Anticipate problem output. Interface must put danger marks where it anticipates catastrophe. Reader is thus forewarned to take special care. Some of the danger spots

may be present in the lexical databases while some may get generated because of interaction of two or more phenomena.

5. Provide on-line help. When the user has some difficulty, the interface should provide him with help. The interface should try to identify the source of difficulty and provide the most relevant information. For this it will be necessary for the interface to keep track of what the user knows and what the output has been produced so far.

6. Provide user friendly access to knowledge base. When all else fails to solve a difficulty, the user will try to solve it on his own. It will be necessary, therefore, to provide access to historical, etymological, linguistic, and cultural knowledge.

4.5.2 Issues in Design

First, there is the issue of focussing user attention on relevant part of the screen and to presenting multiple contexts if needed. These are addressed today by pop-up windows and pull-down menus.

For example, the extended sense of khaa (to eat) is shown using backquote mark khaa‘ (eat‘) in anusaraka output. The sense of this word might be shown as

khaa‘ (eat‘)			
piinaa (drink)	dhumra_paana_kara (smoke)	khaa (eat)	cuuma (kiss)

The second issue in design is how much information should be presented in the first instance. If too much detail is presented, size will become huge and the reader will have difficulty in understanding the meaning. If too little detail is presented, the user might be misled or will have to seek help repeatedly. The correct amount of detail is to be presented, for successful operation. Somewhat like focussing a telescope.

4.5.3 Intelligent User Interface

This issue relates to user model and task model. User model is needed so that the machine is able to provide an appropriate response or help to the user when needed. The user model, in general, includes not only what the user knows but also his plans and goals. The latter are also related to task model. The interface will have to facilitate the following functions:

(a) Remember what it is conveyed to the user earlier (it should not normally repeat things), what the user typically forgets (it should be constantly reminded at appropriate times, till the user learns them), and what the user knows. All these will be part of the user model.

(b) Infer the most likely needs of the user in the given situation.

(c) Present the information in an appropriate form

The output produced by anusaraka can be analysed and the resulting analysis compared with the original analysis of the source sentence. If the analysis of a target language string has a reading not contained in the source string, it means that a new sense is being introduced while rendering it in the target string. Take for instance the sentence (4.2) that we have seen earlier.

(4.2) K: kaanpura nanageV hoVsa uuru.
Kanpur me-to new attach/town
(Kanpur is a new town me.)
H: kaanapura mere liye nayaa shahara hei.
@H: kaanapura mujhe nayaa lagaa/jukaa[gaazva].

An analysis of "lagaa" using a Hindi morph shows that it is ambiguous between: lagaa-(imperative) and laga-(past). The second sense is not contained in the original Kannada.

Since the reader knows the target language, he would find the same ambiguity. In other words, this gives us a method of modelling how the user would interpret the output, and what misconceptions he might be proned to.

In case a new sense is introduced in the output, the system could either try to use (a) additional notation or (b) produce a different target language string. Note that to implement all this in Kannada-Hindi anusaraka, for example, we will also have to incorporate a Hindi analyzer.

Chapter 5

Conclusions

We have argued that it is possible to overcome the language barrier in India using anusaraka. Anusaraka tries to take advantage of the relative strengths of the computer and the human reader; the computer takes the language load and leaves the world knowledge load to the reader. It is particularly effective when the languages are close, as is the case with Indian languages. Keeping in line with the anusaraka philosophy, it bridges the gap between languages by chosing the most appropriate or nearest construction available in the target language together with suitable additional notation.

It needs to be re-stated that even without a sentential parser, the anusaraka for Indian languages delivers something practical today. More detailed sentential or text analysis requires preparation of karaka charts and other lexical databases. These can be gradually built and incorporated in the system. Thus, the system is designed to grow modularly.

Now, we will discuss how anusaraka can be useful as a measuring tool the linguists. Anusaraka can be used to get accurate and large amounts of cross linguistic data from information theoretic viewpoint. Finally, future directions of work are indicated.

5.1 Anusaraka as a Measuring Device for the Linguist

Here, we will focus on another aspect of anusaraka—namely as an instrument for research on language. It allows a linguist or language scholar to study a language (say S) in terms of his own language (say T). He will be able to see constructions, function words, content words "as they are used" in language S. Since anusaraka extracts information that is actually coded in the source text or strings and renders it in language T' (a language close to T), the focus will be on information and how it is coded in the respective languages. As the anusaraka maintains a clean separation between language knowledge and world knowledge, which gets reflected in actual coding of information in a language string, as opposed to its interpretation using world knowledge, it allows the linguist to focus on the former.

The role that anusaraka can play in language research can be compared to that played by the telescope in astronomy. With naked eyes, one could see only a limited number of features and not always accurately, in the stars and the planets. The telescope allowed much more extensive and accurate amounts of data to be obtained. The theory making that followed gave birth to almost a new science of astronomy. In language research, the gloss that is provided for the strings of language S under study, correspond to "naked eye data of astronomy". First, it is painstakingly produced by sitting and working with a speaker of language S. Thus, it places limits on the amount of data that can be looked at (from half a dozen strings to a few hundred at most). Second, such glosses, since they are produced by humans, often tend to bring in and mix world knowledge. Human beings make strong use of world knowledge, in other words, they interpret the string, making use of not only what is explicitly coded but also what is "implied". This brings in inaccuracies in the data. Third, certain kinds of data by their very nature require large studies. Such kinds of data are missed or deliberately ignored. An example is word senses: For a given word in S its senses and the corresponding alternative lexical items in the target language are ignored

keeping only the appropriate word (sense) in the gloss.

The availability of anusaraka will change all this. It will allow large number of source language strings (say thousands and even possibly millions of sentences from corpora) to be studied by a language researcher in terms of the language (T) that he knows. The output produced by anusaraka will be free of "interpretation" based on world knowledge because it will be mechanically produced (where world knowledge can be set to empty). It will also allow exploration of ideas relating to lexical items and senses.

It should be mentioned that the anusaraka could be "focussed" to look at some selected language phenomena or constructions. By the "turn" of some appropriate knobs, it could look only at the selected phenomena in a given corpora. The number of alternatives it shows for say a word, can also be controlled as desired. Like any sophisticated measuring instrument, it can be provided various input parameters that can be controlled by the researcher who is operating it.

The study of Kannada-Hindi anusaraka indicates that from the point of view of information coding, only a small number of differences between South Indian languages and Hindi are important. The most striking among them is the "jo" construction discussed earlier in Chap. 3. Here we discuss the "naa" construction.

The nominal participial phrases in Kannada do not convey information about karaka relations between the main verb and the participle. For example, tinnalu (to eat) in the following two sentences does not mark the relationship this participle has with the main verb.

```
K:  raama  tiMDi      tinnalu  hoTeVlligeV    hodanu
    Rama   breakfast to eat    hotel-acc.     went
   (Rama  went to the hotel to eat breakfast.)
```

@H: raama naashtaa khaanaa_* hoTala ko gayaa

H: raama naashtaa khaane ke liye hoTala gayaa

K: magu tiMDi tinnalu taayigeV saMtoSa aayitu

child breakfast to-eat mother happy be-past

(The mother was happy with the child eating the breakfast.)

@H: baccaa naashtaa khaanaa_* maaz ko santoSa huaa

H: bacce ke naashtaa khaane se maaz ko santoSa huaa

Based on our world knowledge, we can conclude that in the first sentence, the participle 'tinnalu' (to eat) is sampradana of the main verb 'hodanu' (went) whereas in the second sentence it has an entirely different relationship with its main verb. Note that in anusaraka Hindi, a '*' shows that the postposition marker cannot be decided because the semantic relation information is absent in the source sentence. (Based on world knowledge, the first '*' can be substituted by 'ke liye' and the second by 'se'.)

Similar are the following two sentences:

K: raamanigeV odalu baruttadeV

@H: raama ko paDhanaa_* aataa hei

Rama dat. to-read come-pres

(Rama knows how to read.)

K: maLeV baralu keVreV tuMbitu

@H: baarish aanaa_* taalaaba bharaa

rain to-come pond filled

(The pond filled with the rainfall.)

This phenomenon needs further study. For example not all postposition markers can replace '*' in the output.

5.2 Future Potential

Anusaraka can be viewed from the following different points of view. These suggest its future potential.

1. Anusaraka as an evolutionary system: Anusaraka delivers something practical today without waiting for several years and has the potential to keep pace with developments in technology; the work to be done for building it, needs to be done in any case for high-quality fully automatic machine translation systems; and its availability will help in accelerating the work towards development of machine translation systems of the future.

2. A practical approach to develop intermediate language (or interlingua) for a group of languages: Designers can get a first hand view of various source language phenomena in terms of the language the designer knows.

3. A modular system: By adding suitable modules, it can be converted into—

   - A device to understand source language text.
   - Human aided machine translation system.
   - Fully automated high quality domain specific machine translation system.

4. A way to factorize language knowledge part from world knowledge part: Anusaraka suggests a clean way to separate the language knowledge from world knowledge, and how to use them in construction of a system in a systematic way.

5.3 Future Work

This thesis opens up many avenues for future research. At one end of the spectrum is further work in refinement of the present system. At the other end is the development of a science of language. We discuss these in more detail below.

1. A more refined device: The present Kannada-Hindi anusaraka can be refined further. Refinements can come from several sources. First, more detailed work can be done on lexicon. Nuclear senses and related senses of Kannada words can be studied in greater detail. This work ought to be undertaken with corpora of actual language use. Derivation processes for deriving words from the root can also be made use of in identifying and relating different senses of a word. As discussed in the thesis, identification of the nuclear sense in the source language can help in obtaining an appropriate substitutable word in the target language. Study of such mappings between words in the source language and the conceptual sense space on one hand, and the sense space of target language words on the other hand, can yield new insights into lexical items and their semantics. Perhaps there is a hidden topology waiting to be discovered.

   One need not wait for a theory of sense spaces before putting in more information in lexicon. There is a need to provide detailed etymological and historical information about the words in the source language. Even without a comprehensive theory about sense spaces, the reader will be able to use this information while reading anusaraka output. On the contrary, the theory will probably evolve out of research conducted using corpora, detailed dictionaries and the anusaraka.

   Second, a sentential parser can be included in the present Kannada-Hindi anusaraka. This will require preparation of large lexical databases including such things as karaka charts for all the verbs, but this is a one time work. Inclusion of the sentential parser should improve the quality of output, but it is right now not clear whether there will

be any substantial improvement in case of Kannada-Hindi anusaraka. It should also be mentioned that for the parser to be robust, it should be able to handle ellipses. Concept of usability would ultimately decide whether the increase in complexity and lack of transparency will be balanced by the improvement in quality of output.

2. Better user interface: Much more work is needed to improve the user interface. This is the place where AI can make major contributions. Important will be, issues in user modelling and task modelling. There has been considerable discussion on this in Chapter 4.

   With suitable interfaces, anusaraka can become HAMT (Human Aided Machine Translation) or a translator's aid. Such interfaces need to be built and tried out in practice. It should be remembered that the requirements of the translator's aid vs. a common man's aid are different. The former is a translator and will use the device only if it increases his productivity. Consequently, it should help him in that part of the translation task where he spends large amounts of time. Also needed would be good post-editors.

3. Language study: It has already been discussed in this chapter how anusaraka can become "the telescope for the linguist". It will allow him to collect and analyse large amounts of data and more accurately. Such studies need to be undertaken among various Indian languages using anusaraka. This will lead to new comparative and contrastive data and theories. There is another important aspect of such studies. As anusaraka preserves information, it will push information based theories to centre stage of language study, which are needed for NLP. Though communication (or transfer of information) is the primary function of language, not enough attention is paid to this primary function in Chomskyan model. Once anusaraka starts getting used like a measuring instrument with large corpora, it can give linguistics a scientific dimension.

As lord Kelvin said, the difference between science and non-science is measurement.

4. English to Indian languages anusaraka: This is clearly a very important area of work because of its practical utility. It is of theoretical interest too, because one has to deal with two different kinds of languages: fixed word order and free word order, relatively speaking.

   An English to Hindi anusaraka, for example, will surely make use of a sentential parser for English. The notions of subject, object etc., will get mapped on to postposition markers in Hindi. Word order can also be suitably changed. These issues need further study. A Paninian parser for English, in which subject, object etc. are nothing but generalized vibhakti. This notion might have certain advantages as the parser is already working with vibhaktis. Changing word order will once again bring issues of transparency and usability. Also of interest are word senses in English and how they relate to words in Indian languages. This will have important implications for size of output text when compared with size of input text. This in turn determines the frequency of mild catastrophes.

   It is important to mention that the experience gained in building anusarakas among Indian languages is a necessary pre-requisite for building anusaraka for English—an Indian language. This experience relates to notions discussed at length in this thesis, such as, language bridges, sense spaces, tradeoff between mild and serious catastrophe, notation for controlling size of output, interface design, and above all, usability.

Bibliography

[1] Acharya, P. V. *Padartha Chintamani.* Dattatreya Prakashana, Mangalore, 1991.

[2] Adler, Paul S. and Terry A. Winograd, editors. *USABILITY: Turning Technologies into Tools.* Oxford University Press, Inc., 1992.

[3] Barton, G Edward, Berwick C. Robert, and Eric Sven Ristad. *Computational Complexity and Natural Language.* The MIT Press Cambridge, 1987.

[4] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. A Computational Framework for Indian Languages. Technical Report TRCS-90-100, Dept. of CSE IIT Kanpur, July 1990b. (Course notes for Intensive Course on NLP for Linguists, Vol. 1).

[5] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. Local Word Grouping and Its Relevance to Indian Languages. In Vijay P. Bhatkar and Kiran M. Rege, editors, *Frontiers in Knowledge Based Computing (Proc. of KBCS90)*, pages 277–296. Narosa Publishing House, New Delhi, 1990c.

[6] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. Computational Linguistics and Its Relation to Linguistics. *International Journal of Dravidian Linguistics*, XXI(2):106–114, June 1992a.

[7] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. LFG, GB and Paninian Frameworks: An NLP Viewpoint. In *Tutorial on NLP at CPAL-2: UNESCO 2nd Regional*

Workshop on Computer Processing of Asian Languages, Dept. of CSE, IIT Kanpur, 12–16 March 1992b. (Also available as TRCS-92-140, Dept. of CSE, IIT Kanpur.).

[8] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. *Course material on A Computational Grammar Based on Paninian Framework.* Indian Society for Technical Education, Ministry of Human Resource Development, New Delhi, October 1993c.

[9] Bharati, Akshar and Rajeev Sangal. Parsing Free Word Order Languages using the Paninian Framework. In *ACL93: Proc. of Annual Meeting of Association for Computational Linguistics.* Association for Computational Linguistics , NY., 1993a.

[10] Bhatta, Shankara D. N. *Kannada Vakyagalu- Antarika Rachane mattu Arthavyavasthe* . Geetha Book House, Mysore, 1978.

[11] Dasgupta, Probal. Computational Grammars for Machine Translation: A Linguistic Perspective. Invited paper for the postponed National Workshop on Technology Support for Indian Languages, I.I.T. Kanpur, unpublished, 1991.

[12] Gazdar, G., E. Kleine, G.K. Pullum, and I.A. Sag. *Generalized Phrase Structure Grammar.* Basil Blackwell, 1985.

[13] Geetha, K. *Subsystems of Principles: A Study of Universals Based on Tamil Syntax.* PhD thesis, Dept. of HSS, IIT Kanpur, 1985.

[14] Jain, Abhilasha. *Theta-Roles in Syntax: A Theory of Some Dependent Elements of Hindi.* PhD thesis, Dept. of HSS, IIT Kanpur, 1990.

[15] Joshi, Aravind K. Tree Adjoining Grammars: How much context-sensitivity is required to provide reasonable structural description. In Dowty D., Karttunen L., and Zwicky A., editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge, UK., 1985.

[16] Joshi, Arvind K. Processing Crossed and Nested Dependencies: An Automaton Perspective on the Psycholinguistic Results. *Language and Cognitive Processes*, 5(1), 1990.

[17] Kittel, F. *A Grammar of the Kannada Language.* Asian Educational Services, New Delhi, 1982.

[18] Levinson, S. *Pragmatics.* Cambridge University Press, 1983.

[19] Mosaic correspondent. The Science of Learning Math and Science. *MOSAIC*, 23(2), Summer 1992.

[20] Mysale, J. D., editor. *Adarsha Kosha Kannada-Hindi.* Ramashraya Book Depot., Dharwad, Karnataka, 1957.

[21] Pullum, Geoffrey K. Foot loose and Context-free . In Jack Kulas, James H. Fetzer, and Terry L. Rankin, editors, *Philosophy, Language and Artificial Intelligence*, pages 69–78. Kluwer Academic Publishers Netherlands, 1988.

[22] Schiffman, Harold. A Reference Grammar of Spoken Kannada. Technical Report OE,G00-78-01861, Dept. of Asian Languages, University of Washington, Seattle, 1978.

[23] Shieber, Stuart M. Separating Linguistic Analysis from Linguistic Theories . In U. Reyle and C. Rohrer, editors, *Natural Language Parsing and Linguistic Theories.* D. Reidel, Dordrecht, 1988.

[24] Tomita, Masaru. *Efficient parsing for natural language: a fast algorithm for practical systems.* Kluwer Academic Publishers Netherlands, 1986.

Appendix A

Minor Bridges between Kannada and Hindi

1. ko and ko'.

   Kannada has distinct accusative and dative case markers and these map to *ko* in Hindi, as Hindi has the same case marker *ko* for both accusative and dative cases. Anusaraka Hindi distinguishes these two case markers as:

   (a) *ko* as accusative case marker

   (b) *ko'* as dative case marker

   K: raama pustakavannu siiteVgeV koVTTanu.
      Rama Book-acc. Sita-dat. give-past
   @H: raama pustaka ko siitaa ko' diyaa.
   H: raama ne pustaka siitaa ko dii.

2. se and kii_apekSaa.

   Kannada has distinct instrumental/ablative and comparative markers and these map to *se* in Hindi, as Hindi has the same case marker *se* for the above cases. Anusaraka Hindi distinguishes these as:

(a) *se* as instrumental/ablative case marker

(b) *kii_apekSaa* as comparative marker

K: raama camacadiMda uuTa maaDidanu.
Rama spoon-inst. meal do-past
(Rama ate the meal with a spoon.)
@H: raama cammaca se bhojana kiyaa.

K: raama meisuuriniMda beVMgaluurigeV baMdanu.
Rama Mysore-abl. Bangalore-dat. come-past
(Rama came from Mysore to Bangalore.)
@H: raama meisuura se beMgaluura ko' aayaa.

K: raamanigiMta kRSNa ceVnnaagi iddaaneV.
Rama-comp. Krishna fair be-pres
(Krishna is fairer than Rama.)
@H: raama kii apekSaa kRSNa acchii_taraha hei.

3. Negation.

Kannada has two distinct negation words, i.e., *'alla'* and *'illa'*. *'alla'* is used for identity negation and *'illa'* is used for existential negation. These two negation words map to a single word *nahiiM* in Hindi. But anusaraka Hindi distinguishes these two negation words as:

(a) *alla* is mapped to *nahiiM'*

(a) *illa* is mapped to *nahiiM_hei*

K: avanu raama alla, mohana.

he Rama is-not Mohana

(He is not Rama, but Mohana.)

@H: vaha'{pu.} raama nahiiM' , mohana.

K: avanu uurinalli illa.

he town-in is-not

(He is not in the town.)

@H: vaha'{pu.} gaazva meM nahiiM_hei

4. Reporting or narrating element 'aMteV'.

Kannada (like other Dravidian languages) has a narrating element *'aMteV'* which occurs as a sentence suffix. It does not have an equivalent in Hindi. Consider the following sentence:

K: oVMdu uurinalli oVbba raaja iddanaMteV.

[oVMdu uurinalli oVbba raaja iddanu] aMteV.

one town-in one King be-past

(It is said that there was a King in a town.)

@H: eka gaazva meM eka raajaa rahaa eisaa kahate heiM.

Anusaraka Hindi has coined a new phrase *'eisaa_kahate_heiM'* for handling the above sentence suffix.

5. yaha_ and yaha' / vaha_ and vaha'.

Kannada has a proximate demonstrative adjective *ii* and a corresponding neuter demonstrative pronoun *idu*. Both these words map to the same word *yaha* in Hindi. Anusaraka distinguishes these two as:

(a) ii is mapped to yaha_

(b) idu is mapped to yaha'{na.}

Here na. stand for napuMsakaliMga (neuter gender).

```
K:  ii   pustaka nanageV daariyalli sikkitu.
    this book    me-to   road-in    get-past
    (I got this book in the road.)
@H: yaha_ pustaka mujhe' saDaka meM milaa.

K:  idu  pustaka alla.
    this book    is-not
    (This is not a book.)
@H: yaha'{na.} pustaka nahiiM'
```

Similarly the distant demonstrative adjective *aa* and the corresponding neuter demonstrative pronoun *adu* are mapped to *vaha_* and *vaha'{na.}* respectively in anusaraka Hindi.

6. Additional information in anusaraka Hindi.

There is a three way gender distinction among Kannada pronouns viz., *'avanu'*, *'avaLu'*, and *'adu'* corresponding to he, she and it of English. In Hindi, all these three words map to a single word *'vaha'* which does not contain gender information. However, anusaraka faithfully reproduces the gender information that was present in the source language as follows:

(a.) avanu is mapped to vaha'{pu.}

(b.) avaLu is mapped to vaha'{strii.}

(c.) adu is mapped to vaha'{na.}

Here pu., strii., and na. stand for pulliMga (masculine gender), striiliMga (femanine gender), and napuMsakaliMga (neuter gender) respectively.

In Kannada, there are two distinct words *'neVnneV'* and *'naaLeV'* corresponding to English yesterday and tomorrow. These map to a single word *'kala'* in Hindi. To convey the additional information that was present in Kannada, anusaraka handles them as shown below:

(a.) neVnneV is mapped to kala{biitaa_huaa}

(b.) naaLeV is mapped to kala{aanevaalaa}

Appendix B

Some Common Misconceptions about Anusaraka

There are some common misconceptions about anusaraka. We describe them below.

- Anusaraka is a "rough" translation system: Strictly speaking, this is not true. Conceptually, anusaraka is different from translation. It provides exactly the information contained in the source text, while translation involves interpretation of the source text before expressing it in the target language. In case of legal documents, therefore, translation seldom suffices. Anusaraka on the other hand will only say what is explicitly stated. When one is willing to put in extra effort and time then on-line output of anusaraka can be superior to any translation. Because anusaraka makes full surface information, as well as complete language knowledge available to the reader, if the reader is willing to take some pains, he can get full appreciation of the original text. The effort on the part of a reader can be minimized by:

  1. the proper design of the intelligent user interface;
  2. proper training on the part of reader;
  3. practice in using the system.

- Ad hoc improvements such as partial agreement can improve the performance of the system: This is not correct, because then the user will not be sure about what to expect and what not to expect from the machine. It pays to keep working of the machine simple. Syntactic sugar is to be added at personal risk. It may be injurious sometime. Ad hoc improvements which work for one text might cause grave problems with other texts.

- Anusaraka, in principle, is against having a sentential parser: Anusaraka is not against sentential parsers. A sentential parser may be included depending on the availability of technology and linguistic databases. As the Indian languages are close, vibhakti mapping is quite effective. Therefore, anusaraka for Indian languages can be built without waiting for the parser to be available. Later, when large computational lexical databases are available, parser can be incorporated. In fact, for an anusaraka from English to Indian languages, a parser will be absolutely necessary.

Appendix C

Vibhakti Chart

CASES	Kannada case endings	Hindi case endings
nominative	ϕ,nu	ϕ
accusative	nannu	ko
instrumental	niMda	se
dative	nigeV	ko‘
ablative	nadeVseVyiMda	ke_kaaraNa
genitive	na	kaa
locative	nalli,noVLu	meM
vocative	ne,aa	0‘,e
benificial	nigaagi,nigoskara	ke_liye
comitative	noVMdigeV,noVDaneV	ke_saatha
goal	navareVgeV,natanaka	ke_taka
allative	natta	ke_tarapha
causal	niMdaagi	ke_kaaraNa
source	nallina	meM_kaa
addessive	nalligeV	ke_paasa

PARTICLES OF SPECIAL SYNTACTIC FUNCTIONS

function	Kannada Noun ending	Hindi Noun ending
comparative1	naMteV	ke_jeise
comparative2	nigiMta	ki_apekSaa
comparative3	naMtaha,naMtha	jeisaa
equative	naStu	ke_utanaa
topic	naMtuu	to
predicative_nom.	naddu,nadu	vaalaa
dubitative	no	yaa
interrogative	ne,naa,no	kyaa
emphatic1	ne	hii
emphatic2	nuu	bhii

Appendix D

Local Word Grouping (LWG) for Kannada

Noun Grouping:

Kannada	Hindi	
	Before lwg	**After lwg**
raamana haageV	raama_kaa usa_prakaara	raama_ke_jeise
raamana tanaka	raama kaa taka	raama taka
raamana vareVgeV	raama kaa taka	raama taka
raamana saluvaagi	raama kaa ke_lie	raama_ke_lie
raamana maTTigeV	raama kaa paryaMta	raama_ke_stara
hattu halavu	dasa kaii	dasiyoM
tiMDi tiirtha	naashtaa tiirtha	naashtaa_paanii

Verb Grouping:

Kannada	Hindi	
	Before lwg	**After lwg**
tiMda meleV	khaayaa_huaa uupara	khaane_ke_baada
tiMda naMtara	khaayaa_huaa anaMtara	khaane_ke_baada
tiMda baLika	khaayaa_huaa baada	khaane_ke_baada
tiMda kuuDale	khaayaa_huaa turaMta_hii	khaate_hii
tiMda kuuDaleV	khaayaa_huaa turaMta	khaate_hii
tiMda oVDaneV	khaayaa_huaa jhaTa	khaate_hii
tiMda hoVratu	khaayaa_huaa atirikta	khaaye binaa
tinnuva vareVgeV	khaanaa_vaalaa taka	khaane_taka
tinnuva tanaka	khaanaa_vaalaa taka	khaane_taka
tinnuva saluvaagi	khaanaa_vaalaa ke_lie	khaane_ke_lie
tinnada meleV	nahiiM_khaane_vaalaa uupara	nahiiM khaanaa_hei_to
tiMdiraluu saaku	khaanaa_para_bhii paryaapta	khaayaa bhii_hogaa
tiMdiraluu bahudu	khaanaa_para_bhii hogaa	khaayaa bhii_hogaa
tinnale illa	khaa sakataa hei_kyaa nahiiM_hei	khaayaa_hii_nahiiM
tinnuttaa illa	khaataa hue nahiiM_hei	nahiiM_khaa_rahaa_hei
tinnuttitto eno	khaataa_thaa kyaa[to]	shaayada_khaataa_thaa
tinnuvano illavo	khaayegaa_kyaa_hei_ki_nahiiM	khaayegaa ki_nahiiM
tiMdano illavo	khaayaa_kyaa hei_ki_nahiiM	khaayaa ki_nahiiM
tiMditte hoVratu	khaataa_thaa_atirikta	khaayaa hii_maatra
tiMde tinnuvanu	khaa_kara_hii_khaayegaa	jaruura_khaayegaa
tiMde tinnuttaaneV	khaa_kara_hii_khaataa_hei	khaataa_hii_khaataa_hei
tinnabekaagi baMditu	khaanaa_eisaa_aayaa	khaanaa_padaa
tiMdidduu tiMdidde	khaa_kara_bhii_khaayaa_huaa	bahuta_khaayaa
tiMdu hoguttittu	khaa_kara_jaataa_[thaa]	khaa jaataa_thaa

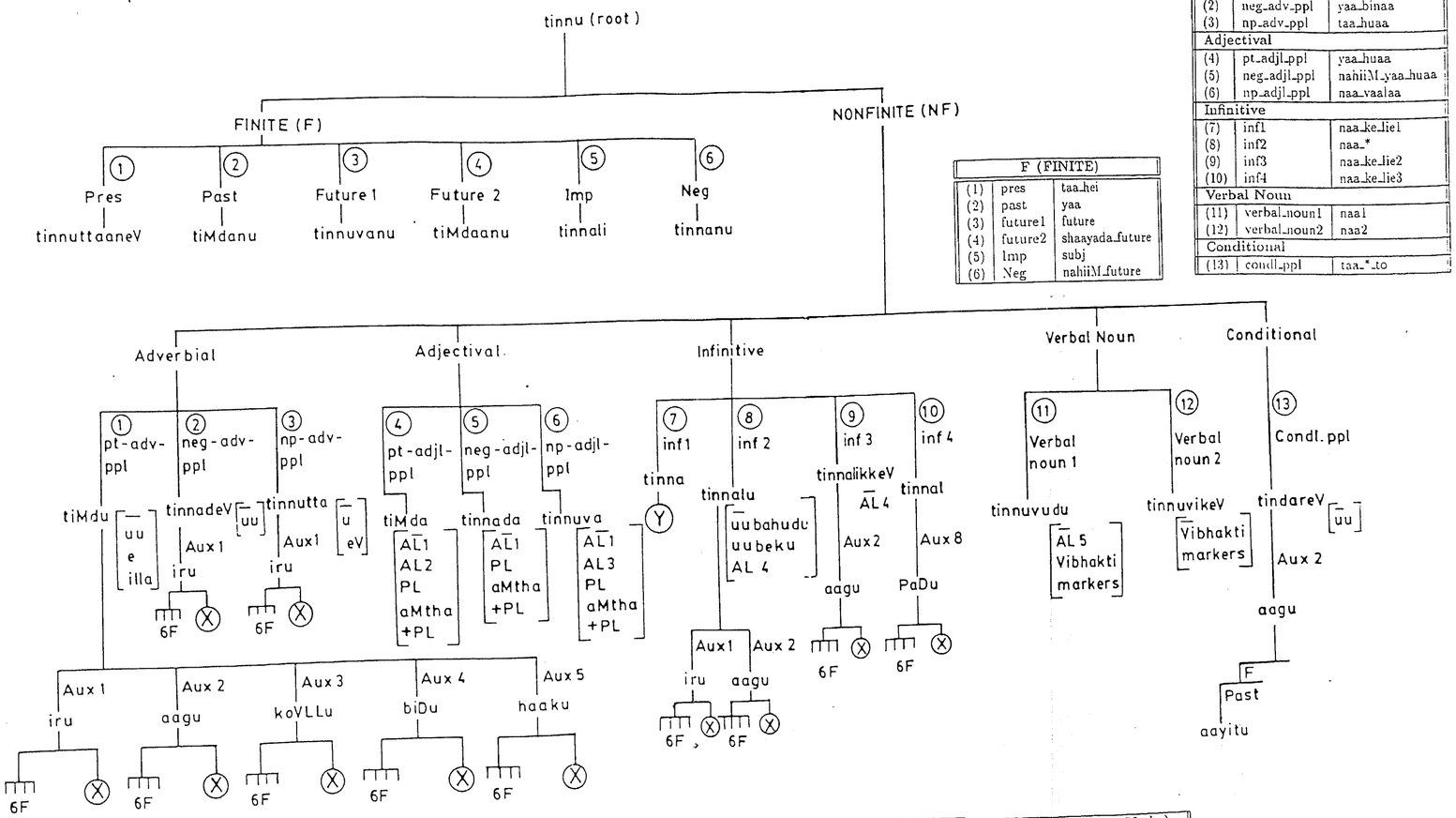
Sample-2

NUSARAKA :: Kannada	Hindi
1. ಅದೇ ಶಿಕ್ಷೆ.	<1> वही सजा.
2. ಗಾಂಧೀಜಿ ಸೇವಾ ಗ್ರಾಮದಲ್ಲಿ ಇದ್ದ ದಿನಗಳಲ್ಲಿ.	<2> गांधीजी सेवा ग्राम में रहा हुआ दिन.
3. ಅವರ ಅನೇಕ ಶಿಷ್ಯರಲ್ಲಿ ಅಕ್ಬರ ಎನ್ನುವ ಒಬ್ಬನು ಸಹ ಇದ್ದನು.	<3> उनके[-न.] अनेक शिष्यों में अक्बर कहना वाला एक व्यक्ति भी रहा.
4. ಅವನು ಒಮ್ಮೆ ದಿನ ಗಾಂಧೀಜಿಯ ಕೋಣೆಯನ್ನು ಶುಚಿಗೊಳಿಸುತ್ತಿರುವಾಗ ಮೇಜಿನ ಮೇಲೆ ಇರಿಸಿದ್ದ ಮೂರು ಮಂಗಗಳ ಮಜಿನ ಗೊಮ್ಮೆ ಜಾರಿ ಕೆಳಗೆ ಬಿದ್ದು ಒಡೆದು ಹೋಯಿತು.	<4> वह'[पु.] एक[मिल] दिन गांधीजी का कमरा को साफ करता हुआ समय मेज का ऊपर रखा हुआ तीन बंदरों का कहुँच का गुडिया फिसल कर नीचे गिर कर टूट गया.
5. ಅಕ್ಬರನ ಮನಸ್ಸಿನಲ್ಲಿ ಭಯ ಆವರಿಸಿತು.	<5> अक्बर का मन में भय घेरा.
6. "ಆ ಮೂರು ಮಂಗಗಳ ಗೊಮ್ಮೆ ಎಂದರೆ ಬಾಪೂಜಿಗೆ ಬಹಳ ಇಷ್ಟ.	<6> "वह तीन बंदरों का गुडिया माने[कहना से] बापूजी को' बहूत पसंद.
7. ಅದು ಒಡೆದು ಹೋಗಿದೆ ಎಂದರೆ ಅವರು ತುಂಬಾ ಕೋಪಿಸಿಕೊಳ್ಳುವರು, ಏನು ಮಾಡಲಿ ? ಏನು ಮಾಡಲಿ ?" ಎಂದು ಯೋಚಿಸಿ ಕೊನೆಗೆ ನಿಜವನ್ನು ಏ ಹೇಳುವುದೆಂದು ತೀರ್ಮಾನ ತೆಗೆದುಕೊಂಡನು.	<7> वह'[न.] टूट गया है माने[कहना से] वे[-न.] [खरोंच] बहूत गुस्सा करेगा, क्या करे? क्या करे? " ऐसा[कह कर/किस दिन] सोच कर [सोचे] अंत को' सत्य को ही कहना चाहिये ऐसा फैसला ले लिया[निकाल लिया].
8. ಸರಿ !.	<8> ठीक[सरके] !.
9. ಬಾಪೂಜಿ ಬಂದರು.	<9> बापूजी आया[बंदरगह].
10. ಅಕ್ಬರ ನಮ್ರನಾಗಿ ನಡೆದುದನ್ನು ಹೇಳುತ್ತ, "ನೀವು ನನಗೆ ಯಾವ ಶಿಕ್ಷೆ ಕೊಟ್ಟರೂ ಸರಿ ಏ, ಅನುಭವಿಸಲು ಸಿದ್ಧ" ಎಂದನು.	<10> अक्बर नम्रता से चला है जौन वह[न.] को कह कर[कहे], "तुम[सहला] मुझे' किस सजा देना पर भी ठीक[सरके] ही, भोगना_* सिद्ध" कहा.
11. ಗಾಂಧೀಜಿ ಥಟ್ಟನೆ ಹುಬ್ಬುಗಂಟಿಕ್ಕಿಕ, "ನಿನಗೆ ತಕ್ಕ ಶಿಕ್ಷೆಯನ್ನು ಏ ಕೊಡುತ್ತೇನೆ ! ಜೀವನ ಪರ್ಯಂತ ಅನುಭವಿಸುವಂಥ ಶಿಕ್ಷೆ" ಎಂದರು.	<11> गांधीजी अचानक भृकुटि तान कर[भृकुटि ताने], "तुझे' योग्य सजा को ही देता है ! जीवन पर्यंत भोगना योग्य सजा " कहा.
12. ಅಕ್ಬರ ಥರಥರ ನಡುಗಲು ಆರಂಭಿಸಿದನು.	<12> अक्बर थरथर कहुँपना_* शुरू किया.
13. ಆಗ ಬಾಪೂಜಿ ಮೆಲ್ಲಗೆ ಹೂಸಿನಗೆ ಬೀರುತ್ತ -"ನಿನಗೆ ಶಿಕ್ಷೆ ಯಾವುದು ಗೊತ್ತೆ ?.	<13> तब बापूजी धीरे झूठी हँसी बहुँटता हुआ -"तुझे' सजा जो[कोई] पता है ?.
14. ಇದು ನೀನು ಹುಡಿದಂತೆ ಜೀವನ ಪರ್ಯಂತ ನೀನು ಸತ್ಯವನ್ನ ಏ ನುಡಿಯಬೇಕು.	<14> आज[चंद्रमा] तू कहा है वैसे जीवन पर्यंत तू सत्य को ही कहना चाहिये.
15. ಅದೇ ನಿನಗೆ ಶಿಕ್ಷೆ !" ಎಂದಾಗ, ಅಕ್ಬರ ಗಾಂಧೀಜಿಯ ಕಾಲಿಗೆ ಬಿದ್ದು ಗಳಗಳನೆ ಅತ್ತುಬಿಟ್ಟನು.	<15> वही तुझे' सजा !" कहा तब, अक्बर गांधीजी का पाव_को'[पहुँव_को'] गिर कर फूट फूट कर [तेजी से] [लगियों का] रो दिया.

Appendix F

Kannada TAM (Tense Aspect Modality) Chart

KANNADA TAM (Tense Aspect Modality) CHART



NF (NONFINITE)		
Adverbial		
(1)	pt_adv_ppl	kara
(2)	neg_adv_ppl	yaa_binaa
(3)	np_adv_ppl	taa_huaa
Adjectival		
(4)	pt_adjl_ppl	yaa_huaa
(5)	neg_adjl_ppl	nahiiM_yaa_huaa
(6)	np_adjl_ppl	naa_vaalaa
Infinitive		
(7)	inf1	naa_ke_lie1
(8)	inf2	naa_*
(9)	inf3	naa_ke_lie2
(10)	inf4	naa_ke_lie3
Verbal Noun		
(11)	verbal_noun1	naa1
(12)	verbal_noun2	naa2
Conditional		
(13)	condl_ppl	taa_*_to

F (FINITE)		
(1)	pres	taa_hei
(2)	past	yaa
(3)	future1	future
(4)	future2	shaayada_future
(5)	Imp	subj
(6)	Neg	nahiiM_future

PL (PRONOUN LIST)	
avanu	vaha{pu.}
avaLu	vaha{stri.}
adu	vaha{na.}

AL1 (AVYAYA LIST1)	
aMteV	veise
haageV	veise
aaga	taba
aagaleV	taba_hii
aagaluu	taba_bhii
aSTu	utanaa
aMtha	veise
aMteVyeV	veise_hii
alli	udhara
saluvaagi	ke_liye

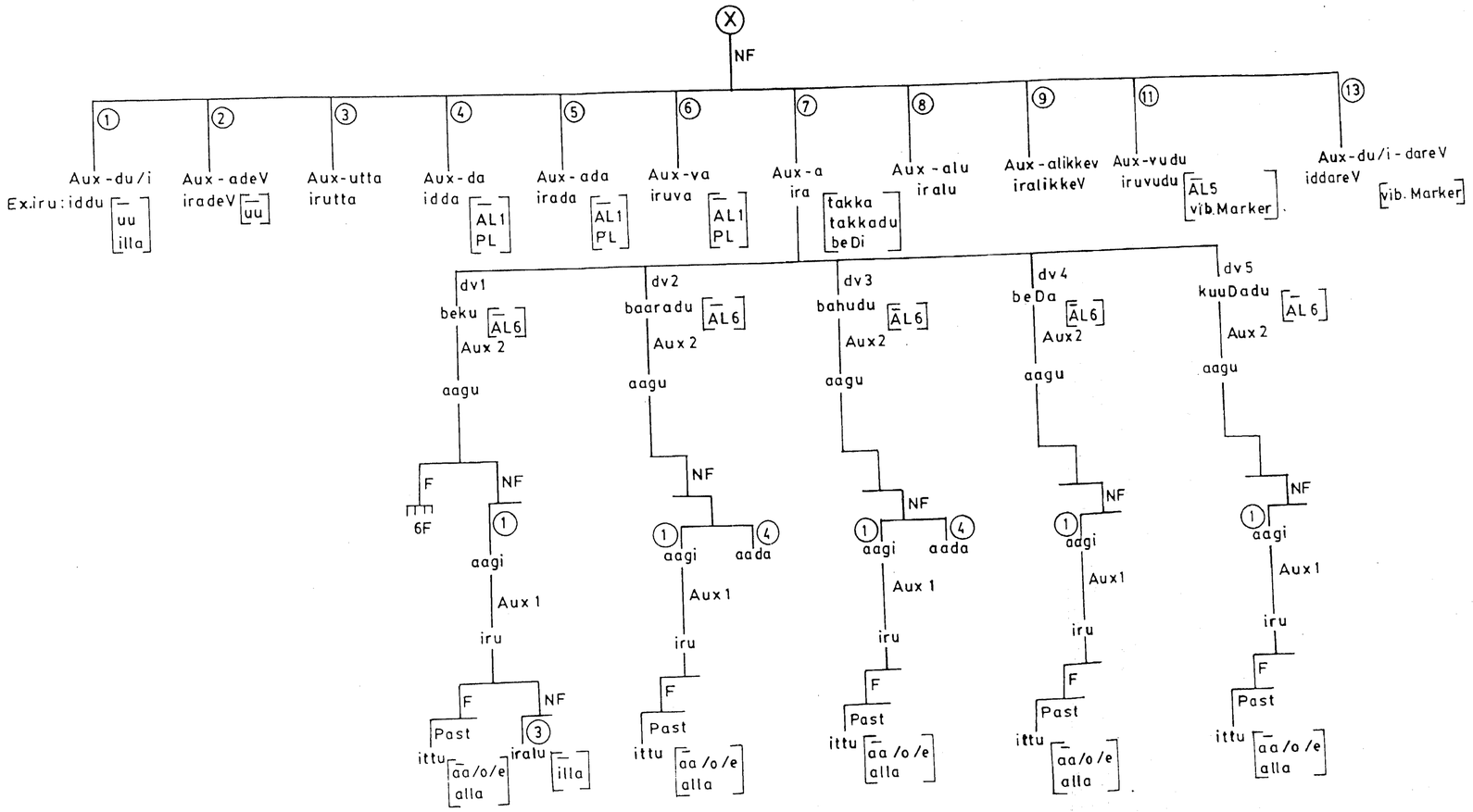
AL2 (AVYAYA LIST2)	
naMtara	baada_meM
meleV	uupara
kuuDaleV	turaMta
oVDaneV	jhaTa
oVDaneVyeV	jhaTa_hii
aMteVlla	veise_saba

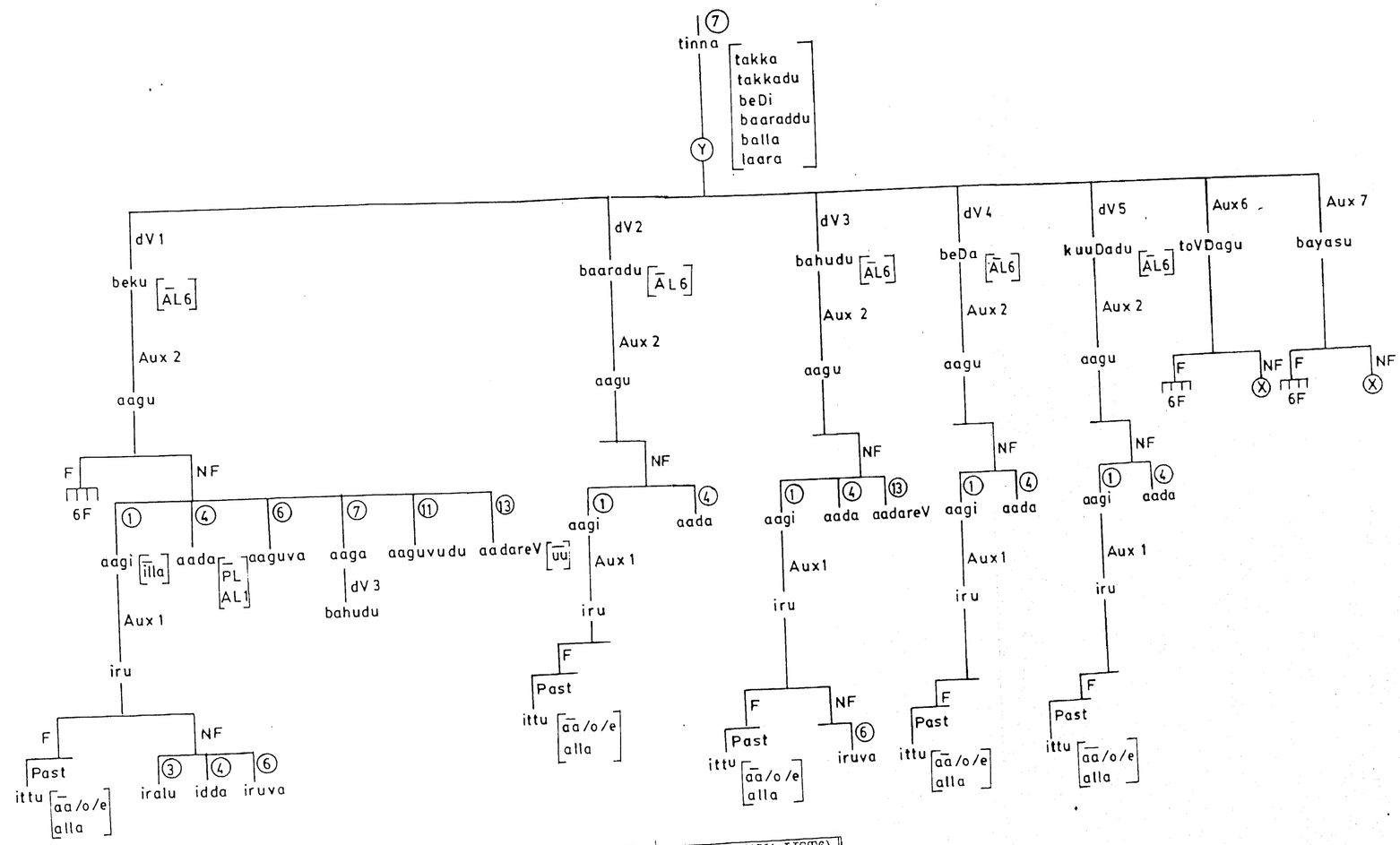
AL3 (AVYAYA LIST3)	
vareVgeV	taka
vareVguu	taka_bhii
tanaka	taka
aSTaralli	utanaa_meM

AL4 (AVYAYA LIST4)	
e	hii
uu	bhii
illa	nahiiM_hei
eVMdu	eisaa

AL5 (AVYAYA LIST5)	
e	hii
uu	bhii
aa	kyaa
o	to
illa	nahiiM_hei
alla	nahiiM
allade V	alaavaa

Aux (Auxiliary Verbs)		
Aux1	iru	rahanaa
Aux2	aagu	honaa
Aux3	koVLLu	lenaa
Aux4	biDu	choDanaa
Aux5	haaku	Daalanaa
Aux6	toVDagu	laganaa
Aux7	bayasu	caahanaa


X
NF
1 Aux-du/i
Ex.iru: iddu [uu illa]
2 Aux-adeV
iradeV [uu]
3 Aux-utta
irutta
4 Aux-da
idda [AL1 PL]
5 Aux-ada
irada [AL1 PL]
6 Aux-va
iruva [AL1 PL]
7 Aux-a
ira [takka takkadu beDi]
8 Aux-alu
iralu
9 Aux-alikkev
iralikkeV
11 Aux-vudu
iruvudu [AL5 vib.Marker]
13 Aux-du/i-dareV
iddareV [vib. Marker]
dv1
beku [AL6]
Aux 2
aagu
F
6F
NF
1
aagi
Aux 1
iru
F
Past
ittu [aa/o/e alla]
NF
3
iralu [illa]
dv2
baaradu [AL6]
Aux 2
aagu
NF
1 aagi 4 aada
Aux 1
iru
F
Past
ittu [aa/o/e alla]
dv3
bahudu [AL6]
Aux2
aagu
NF
1 aagi 4 aada
Aux 1
iru
F
Past
ittu [aa/o/e alla]
dv4
beDa [AL6]
Aux2
aagu
NF
1
aagi
Aux1
iru
F
Past
ittu [aa/o/e alla]
dv5
kuuDadu [AL6]
Aux 2
aagu
NF
1
aagi
Aux1
iru
F
Past
ittu [aa/o/e alla]



dv (defective verbs)		
dv1	beku	caahiye
dv2	baaradu	nahiiM_caahiye1
dv3	bahudu	hogaa
dv4	beDa	nahiiM_caahiye2
dv5	kuuDadu	nahiiM_caahiye3

AL6 (AVYAYA LIST6)	
aa	kyaa
eVMba	eisaa
eVMdu	eisaa
aMta	eisaa